UNCLASSIFIED

| |
|---|
| AD NUMBER |
| **AD875085** |
| NEW LIMITATION CHANGE |
| TO<br>**Approved for public release, distribution unlimited** |
| FROM<br>**Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; JUN 1970. Other requests shall be referred to Ait Force Institute of Technology, Wright-Patterson AFB, OH 45433.** |
| AUTHORITY |
| **AFIT ltr, 22 Jul 1971** |

THIS PAGE IS UNCLASSIFIED

# AIR FORCE INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY
UNITED STATES AIR FORCE

RANGE MAXIMIZATION
OF AN
AIR-TO-SURFACE MISSILE

THESIS

GGC/EE/70-5 Albert I. Chatmon

## SCHOOL OF ENGINEERING

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

RANGE MAXIMIZATION
OF AN
AIR-TO-SURFACE MISSILE

THESIS

GGC/EE/70-5 Albert I. Chatmon

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Dean of Engineering, Air Force Institute of Technology (AFIT-SE), Wright Patterson AFB, Ohio 45433.

RANGE MAXIMIZATION

OF AN

AIR-TO-SURFACE MISSILE

THESIS

Presented to the Faculty of the School of Engineering of

the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

Albert I. Chatmon, B.S.E.E.

Graduate Guidance and Control

June 1970

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Dean of Engineering, Air Force Institute of Technology (AFIT-SE), Wright-Patterson AFB, Ohio 45433.

# Preface

This presentation is an application of optimal control theory to determine approximations of the angle-of-attack history that maximizes the range of a typical air-to-surface missile trajectory. I attempted to approach the problem as simply and as practically as possible, thus hoping to enhance further the use of optimal control techniques by the practical engineering world.

I wish to express my appreciation to my sponsor, James P. McCarthy, Aerospace Engineer, Aeronautical Systems Division and my advisor, Lt. Col. R. A. Hannen, for their guidance and helpful suggestions in preparing this paper. I would also like to express my appreciation to Capt. Thomas E. Moriarty, Guidance and Control Engineer, Aeronautical Systems Division, for his efforts in getting me interested in the problem, and my ex-neighbor, Lt. Steve Faught, formerly of the Digital Computation Division of the Aeronautical Systems Division, for his help in writing the "Calcomp" computer program used to make the graphs in this presentation.

Albert I. Chatmon

Contents

# List of Figures

## List of Tables

## Symbols

| | | |
|---|---|---|
| $\alpha$ | - | angle of attack; the control variable (deg) |
| $\beta$ | - | quotient of the inner products of the gradient of the $(i+1)^{th}$ iteration and the $i^{th}$ iteration |
| $C_{Do}$ | - | parasite drag coefficient |
| $C_L$ | - | lift curve slope coefficient at trimmed flight conditions (per degree) |
| D | - | drag (pounds) |
| $\Delta t$ | - | time interval of the Runge-Kutta integration formula |
| F | - | thrust (pounds) |
| G | - | gravity constant |
| g | - | gravity ($ft/sec^2$) |
| $\gamma$ | - | flight path angle (deg) |
| H | - | Hamiltonian |
| h | - | altitude of the missile (feet) |
| Hu | - | the gradient |
| $I_{sp}$ | - | specific impulse (seconds) |
| J | - | the objective (ft. or mi.) |
| K | - | first guess of the k-parameter |
| k | - | the parameter used to adjust the control variable per iteration |
| L | - | lift (pounds) |
| $\lambda$ | - | adjoint state |
| M | - | Mach number |
| m | - | mass of the vehicle (slugs) |
| $m_o$ | - | initial mass of the vehicle (slugs) |

| | | |
|---|---|---|
| $m_p$ | - | mass of the missile propellant (slugs) |
| p | - | conjugate gradient search direction |
| q | - | dynamic pressure (slugs/(ft-sec$^2$)) |
| R | - | penalty weighting function (ft/sec or mi/sec) |
| r | - | range of the missile (ft or mi) |
| $R_e$ | - | radius of the earth (ft) |
| $\rho$ | - | density (slugs/ft$^3$) |
| S | - | reference surface area (ft$^2$) |
| t | - | time (seconds) |
| $t_f$ | - | fixed final time (seconds) |
| $\theta$ | - | pitch angle |
| V | - | velocity of the missile (ft/sec) |
| $V_s$ | - | speed of sound (ft/sec) |
| $W_p$ | - | weight of the missile propellant (pounds) |
| $W_o$ | - | initial weight of the missile (pounds) |
| x | - | state variable |

## Abstract

Pontryagin's Maximum Principle, coupled with the conjugate gradient iterative technique, is employed in determining estimates of the two-dimensional, maximum range trajectory of an air-to-surface missile. Angle of attack is used as the control parameter.

The motion of the vehicle is described by four state equations including standard atmospheric data, and lift and drag data obtained from wind tunnel test. In the adjoint equations Lagrangian differentiation formulas are used to approximate the derivatives of lift and drag with respect to velocity and altitude.

Two quadratic cost functions are investigated--one involving a linear range term and the other a quadratic term. Both include a quadratic penalty function involving a weighting function and the square of the control.

# RANGE MAXIMIZATION OF AN AIR-TO-SURFACE MISSILE

## I. Introduction

Maximizing the range of currently operating air-to-surface missiles, thus giving the launch vehicle more escape time and/or distance, is one problem that is of particular interest to the U.S.A.F. Despite the current activity in applying optimal control techniques to missile trajectory problems, little has been done in applying such techniques to practical air-to-surface missile problems. This paper treats the use of such a technique in maximizing the range of a typical, two dimensional, air-to-surface missile trajectory.

The conjugate gradient method is the iterative technique used in this investigation. It is chosen because of the speed in which it converges, and the relative ease in setting up the problem and including control constraints. The major shortcoming of this method is that either the final time must be known or time must be treated as a state variable and another monotonely increasing variable, whose final value is known, used as the independent variable in the state and adjoint state equations. In this problem time and range are the only monotonely increasing variables, and neither end condition is known. Therefore, time is used as the independent variable and "educated guesses" are made

on the final time. These "educated guesses" are actually obtained by first simulating trajectories of constant control histories and then making corrections as more information is obtained from analysis.

The control variable is the angle of attack which determines the thrust vector and the aerodynamic forces acting on the missile, thus determining the range of the trajectory. The angle of attack is a very practical control variable since it can be measured and controlled fairly easily, and its derivatives are not present in the state or adjoint state equations. A control constraint in the form of a penalty function is used in this presentation. This constraint insures that the pitch rate of the vehicle remains small by limiting the angle of attack. Also, limiting the angle of attack insures that the vehicle operates in the linear regions of the lift and drag coefficient curves.

A listing of the digital computer program, including appropriate comments, is included in this paper. To avoid storing the atmospheric and aerodynamic data on tapes, polynomial least squares curve fits of the data are used. This definitely shortens the amount of computer time required for each iteration and reduces the amount of necessary storage. The fourth order Runge-Kutta formula is used to integrate the state and adjoint state equations, while the expanded Simpson's formula is used to integrate

the penalty function.

Of interest in all such missile problems is the evaluation of the adjoint state equations, which requires finding the partial derivatives of lift and drag with respect to the state variables, velocity and altitude. In deriving these derivatives it is assumed that (1) the vehicle is operating in the linear regions of the lift and drag coefficient curves, and (2) the parasite drag coefficient is independent of atmospheric density. Lagrangian five-point differentiation formulas are used to estimate the derivatives of atmospheric and aerodynamic data.

Chapter II treats the non-linear equations of motion of the missile. Chapter III then formulates the optimal control equations using the plant equations of Chapter II and Pontryagin's Maximum Principle. The aerodynamic derivatives used in the adjoint equations are derived in Chapter IV, which is included especially for control engineers. The conjugate gradient technique is presented in Chapter V. The results, conclusions, and recommendations are presented in Chapters VI, VII, and VIII respectively.

## II. Equations of Motion

In forming the equations of motion, the missile is treated as a variable point mass acted upon by thrust, gravity, lift, and drag. Since the range of the missile is comparatively short, the earth is considered flat, and the effects of the earth's rotational rate are neglected.

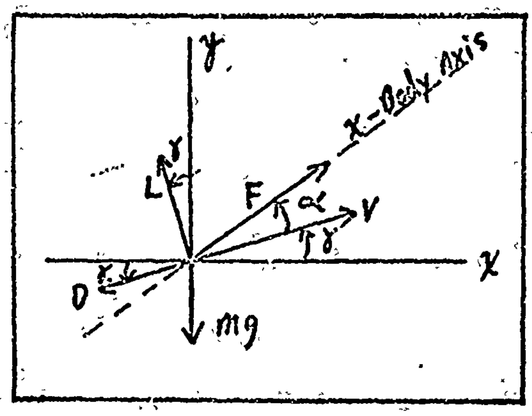


Fig. 1. Free body Diagram of Missile

The reference coordinate system has its x-axis along the surface of the flat earth, its y-axis vertical, and its origin below the point of initial thrust. Fig. 1 depicts the free body diagram of the missile. The fixed x-y coordinate system is the reference coordinate system translated only, such that the missile is located at the origin. The motion measured in the x-y coordinate system is the same as the motion measured in the reference system.

The equations of motion are:

$$
\begin{aligned}
m\dot{V} &= F\cos(\alpha) - D - mg\sin(\gamma) \\
mV\dot{\gamma} &= L - mg\cos(\gamma) + F\sin(\alpha) \\
\dot{h} &= V\sin(\gamma) \\
\dot{r} &= V\cos(\gamma)
\end{aligned}
\qquad (1)
$$

where

$$D = qS\left[C_{D_0} + C_{L_\alpha}\alpha^2\right] \tag{2}$$

$$L = qS\left[C_{L_\alpha}\alpha\right] \tag{3}$$

$$g = \frac{G\, m_E}{\left[R_E + h\right]^2} \tag{4}$$

Appendix B contains a discussion and analysis of Eqs (2) and (3).

The state variables are defined as

$$\overline{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} V \\ \gamma \\ h \\ r \end{bmatrix} \tag{5}$$

where all the variables are functions of time.

### Thrust and Mass

Thrust can be written as a function of time. Fig. 2 is a graph of thrust versus time. The specific impulse $I_{sp}$ (Ref 10: 152) is defined in Eq (6) as

$$I_{sp} = \frac{\text{Thrust (F)}}{\text{weight rate flow of the propellants } (\dot{W})} \tag{6}$$

where

$$\dot{W} = \dot{m}_p\, g \tag{7}$$

Using Eqs (6) and (7), the mass of the missile can be written as

$$m = m_0 - \int_0^t \frac{F}{g_c I_{sp}}\, dt \tag{8}$$

where $M_o$ is the initial value of m — specifically

$$M_o = \frac{W_o}{g_o} \tag{9}$$

where $W_o$ is the initial weight of the rocket. Fig. 3 is a graph of m versus time. For this particular missile $W_o$ is 3000 (lbs) and $I_{sp}$ is 240 (sec).

Equations of thrust and mass as functions of time are located in the computer program (Appendix A).

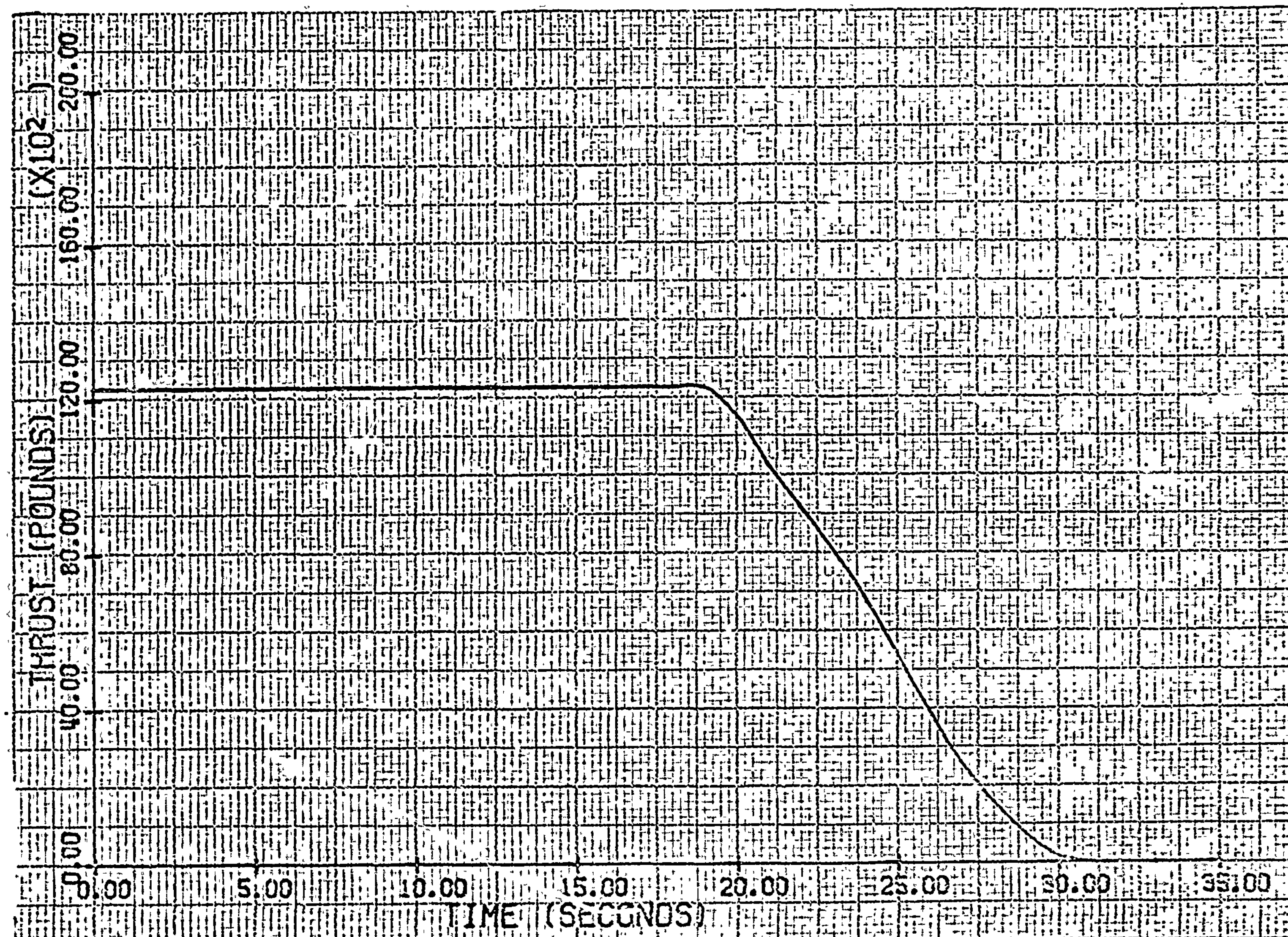

Fig. 2. Thrust Profile of the Missile

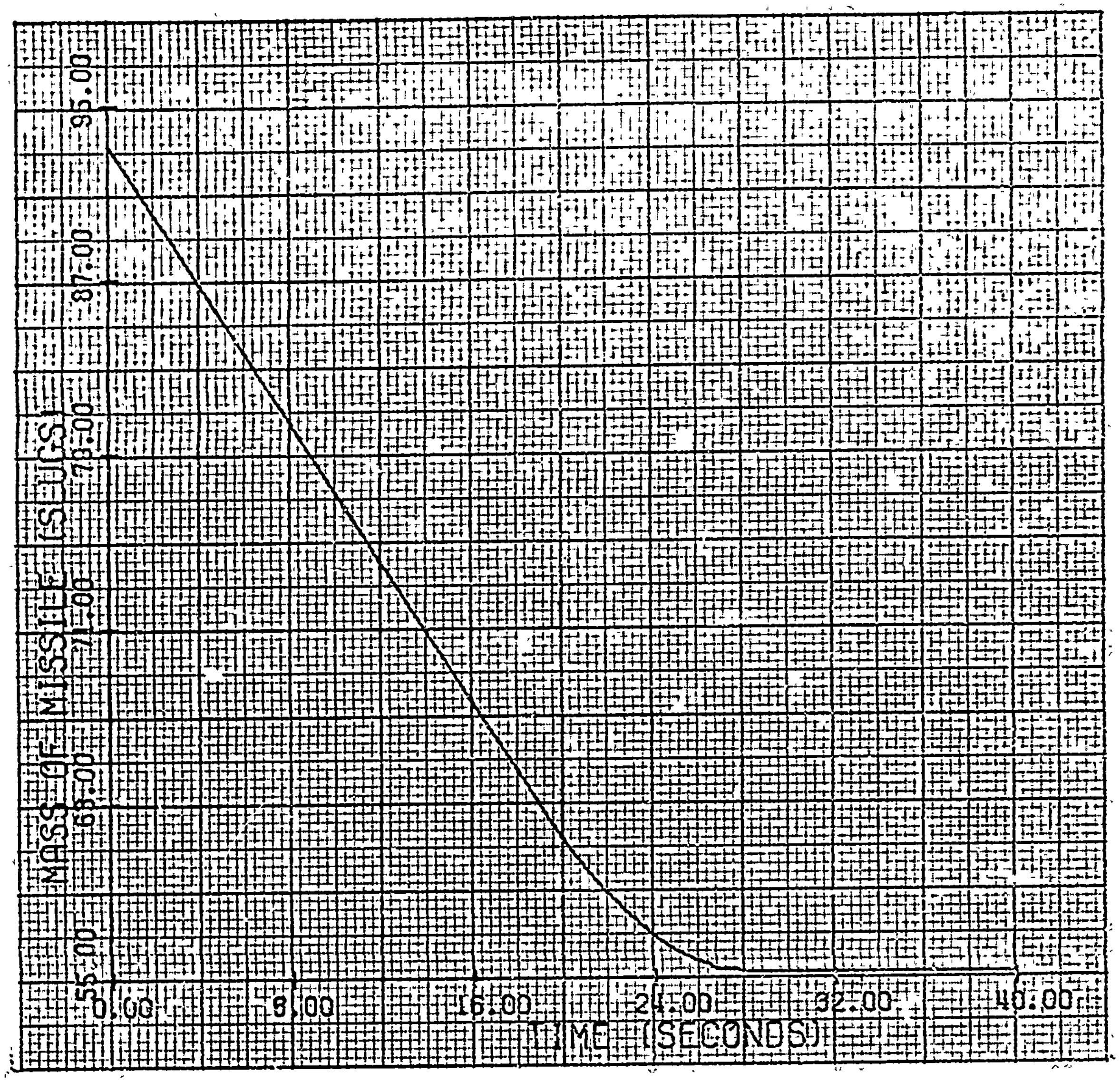


Fig. 3. Mass of the Missile

## III. Formulation of the Optimal Control Problem

### Cost Function

The first consideration is the cost or objective function which must be realized before the optimal control problem can be set up. The most obvious cost functions are

$$J = X_4(t_f)$$

and (10)

$$J = \tfrac{1}{2}\left[X_4(t_f)\right]^2$$

since the objective of this investigation is to maximize range. However, as will be discussed in Chapter V, some penalty on control is needed. Therefore

$$J = X_4(t_f) - R\int_0^t \tfrac{1}{2}\alpha^2\, dt$$

and (11)

$$J = \tfrac{1}{2}\left[X_4(t_f)\right]^2 - R\int_0^t \tfrac{1}{2}\alpha^2\, dt$$

are used.

### Maximum Principle Necessary Conditions

It is desired to maximize J subject to

$$\dot{\bar{X}} = \bar{f}(\bar{x}, \alpha, m, F, D, L, g, t) \qquad (12)$$

where Eq (12) represents Eq (1). For this problem the

necessary conditions of Pontryagin's Maximum Principle are

$$(1)\quad \dot{\bar{x}} = \frac{\partial H}{\partial \bar{\lambda}}$$

$$(2)\quad \dot{\bar{\lambda}} = -\frac{\partial H}{\partial \bar{x}}$$

$$(3)\quad H_u = 0$$

$$(4)\quad \left.\frac{\partial J_1}{\partial \bar{x}}\right|_{t=t_f} = \bar{\lambda}(t_f) \qquad \text{(fixed final time)}$$

where

$$H = -\tfrac{1}{2}\alpha^2 R + \frac{\lambda_1}{m}\left[F\cos(\alpha) - D - mg\sin(X_2)\right] + \frac{\lambda_2}{X_1 m}\left[L + F\sin(\alpha) - mg\cos(X_2)\right] + \lambda_3\left[X_1\sin(X_2)\right] + \lambda_4\left[X_1\cos(X_2)\right]$$

$$\dot{\lambda}_1 = \frac{\lambda_1}{m}\frac{\partial D}{\partial X_1} - \frac{\lambda_2}{m}\left[\frac{\partial(L/X_1)}{\partial X_1} - \frac{F}{X_1^2}\sin(\alpha) + \frac{mg}{X_1^2}\cos(X_2)\right] - \lambda_3\sin(X_2) - \lambda_4\cos(X_2)$$

$$\dot{\lambda}_2 = \lambda_1 g\cos(X_2) - \lambda_2\frac{g}{X_1}\sin(X_2) - \lambda_3 X_1\cos(X_2) + \sin(X_2)$$

$$\dot{\lambda}_3 = \frac{\lambda_1}{m}\frac{\partial D}{\partial X_3} - \frac{\lambda_2}{mX_1}\frac{\partial L}{\partial X_3}$$

$$\dot{\lambda}_4 = 0$$

$$H_u = -R\alpha - \lambda_1\left[\frac{F}{m}\sin(\alpha) + \frac{2qS}{m}C_{L_\alpha}\alpha\right] + \lambda_2\left[\frac{F}{mX_1}\cos(\alpha) + \frac{qS}{mX_1}C_{L_\alpha}\right]$$

$$J_1 = X_4(t_f) \quad \text{or} \quad \tfrac{1}{2}\left[X_4(t_f)\right]^2$$

Using necessary conditions (1), (2), and (3), it can be shown that

$$\frac{dH}{dt} = \text{A CONSTANT} \tag{13}$$

where the Hamiltonian is not a function of time on the optimal trajectory. (Note: in this problem the Hamiltonian is not a function of time where thrust is constant.)

Boundary Conditions

The specified boundary conditions are

$$\begin{aligned} x_1(0) &= 943 \text{ ft./sec.} \\ x_2(0) &= 0 \\ x_3(0) &= 20{,}000 \text{ ft.} \\ x_3(t_f) &= 0 \\ x_4(0) &= 0 \end{aligned} \tag{14}$$

All other final conditions including final time are unspecified. To avoid adding another penalty term to the cost function the only end condition $x_3(t_f) = 0$ is relaxed.

Necessary condition (4) is used to derive the end conditions of the costates:

$$\begin{aligned} \lambda_1(t_f) &= 0 \\ \lambda_2(t_f) &= 0 \\ \lambda_3(t_f) &= 0 \\ \lambda_4(t_f) &= 1 \text{ or } x_4(t_f) \end{aligned} \tag{15}$$

## IV. Aerodynamic Derivatives

In order to solve the costate equations, it is necessary to find the partial derivatives of lift and drag with respect to velocity and altitude. (In the $\lambda_1$ equation lift is divided by velocity.) Using Eqs (2) and (3)

$$\frac{\partial (L/X_1)}{\partial X_1} = qS\left[C_{L\alpha}\frac{\alpha}{X_1^2} + \frac{\alpha}{X_1}\frac{\partial C_{L\alpha}}{\partial X_1}\right]$$

$$\frac{\partial D}{\partial X_1} = qS\left[\frac{\partial C_{D_0}}{\partial X_1} + \alpha^2\frac{\partial C_{L\alpha}}{\partial X_1}\right] + \rho X_1 S C_D$$

$$\frac{\partial L}{\partial X_3} = \frac{qS C_{L\alpha}}{X_1 \rho}\frac{\partial \rho}{\partial X_3} + \frac{qS\alpha}{X_1}\frac{\partial C_{L\alpha}}{\partial X_3} \tag{16}$$

$$\frac{\partial D}{\partial X_3} = qS\left[\frac{\partial C_{D_0}}{\partial X_3} + \alpha^2\frac{\partial C_{L\alpha}}{\partial X_3}\right] + \frac{1}{2}X_1^2 S C_D\frac{\partial \rho}{\partial X_3}$$

where $C_{L\alpha}$ and $C_{Do}$ are functions of Mach number. Since Mach number is a function of velocity and altitude, and atmospheric density is a function of altitude, (Ref 4: 477)

$$\frac{\partial C_{D_0}}{\partial X_1} = \frac{\partial C_{D_0}}{\partial M}\frac{\partial M}{\partial X_1}$$

$$\frac{\partial C_{D_0}}{\partial X_3} = \frac{\partial C_{D_0}}{\partial M}\frac{\partial M}{\partial X_3}$$

$$\frac{\partial C_{L\alpha}}{\partial X_1} = \frac{\partial C_{L\alpha}}{\partial M}\frac{\partial M}{\partial X_1} \tag{17}$$

$$\frac{\partial C_{L\alpha}}{\partial X_3} = \frac{\partial C_{L\alpha}}{\partial M}\frac{\partial M}{\partial X_3}$$

where

$$\frac{\partial C_{D_o}}{\partial \rho} = 0 \tag{18}$$

This assumption implies that parasite drag is directly proportional to atmospheric density if velocity is constant. Wind tunnel and flight tests have shown that this assumption is valid.

By definition Mach number is the ratio of the true air speed to the speed of sound, therefore

$$\frac{\partial M}{\partial X_1} = \frac{1}{V_s}$$

$$\frac{\partial M}{\partial X_3} = -\frac{M}{V_s}\frac{\partial V_s}{\partial X_3} = -\frac{X_1}{V_s^2}\frac{\partial V_s}{\partial X_3} \tag{19}$$

where $V_s$ is a function of altitude. Substituting Eq (19) into Eq (17),

$$\frac{\partial C_{D_o}}{\partial X_1} = \frac{1}{V_s}\frac{\partial C_{D_o}}{\partial M}$$

$$\frac{\partial C_{D_o}}{\partial X_3} = -\left[\frac{\partial C_{D_o}}{\partial M}\right]\left[\frac{X_1}{V_s^2}\right]\left[\frac{\partial V_s}{\partial X_3}\right]$$

$$\frac{\partial C_{L_\alpha}}{\partial X_1} = \frac{1}{V_s}\frac{\partial C_{L_\alpha}}{\partial M} \tag{20}$$

$$\frac{\partial C_{L_\alpha}}{\partial X_3} = -\left[\frac{\partial C_{L_\alpha}}{\partial M}\right]\left[\frac{X_1}{V_s^2}\right]\left[\frac{\partial V_s}{\partial X_3}\right]$$

Curve Fitting Data

A set of equations approximating atmospheric density (Ref 2: 15) are available in the computer program (Appendix A). Fig. 4 is a graph of density versus altitude. $V_s$, $C_{Do}$ and $C_{L\alpha}$ were approximated by using the piecewise, polynomial, least-squares, curve-fit method (Appendix D). The curve fits are graphed in Figs. 5, 6, and 7. The curves with the asterisk are the curve fits; the curves without the asterisk represent the given data points. Data for the speed of sound were obtained from Ref (8: 4). $C_{Do}$ and $C_{L\alpha}$ data were obtained from wind tunnel tests. Table I shows the maximum deviation in percent between the ordinate of the given data points and those of the curve fit. The curve fit polynomial equations are also located in the computer program (Appendix A).

Table I
Errors in Curve Fits

| Dependent Function | Maximum Deviation in Per Cent | Value of the Abscissa |
|---|---|---|
| $V_s$ | .18 | $3.5 \times 10^4$ (ft) |
| $C_L$ | 1.3 | 3 (Mach No.) |
| $C_{Do}$ | 2.1 | 1 (Mach No.) |

Differentiating the Data Curves

In order to solve Eq (20) and then Eq (16), it is necessary to find the partial derivatives of $V_s$ and $\rho$ with respect to altitude, and those of $C_{L\alpha}$ and $C_{Do}$ with respect

to Mach number. An approximation of these derivatives is obtained by using the average of a set of five, five-point, Lagrangian differentiation formulas. A derivation of the formulas is located in Appendix C. The approximations for the errors in the formulas are assumed negligible.

It is now possible to solve Eq (16) if the velocity and altitude of the vehicle are known.

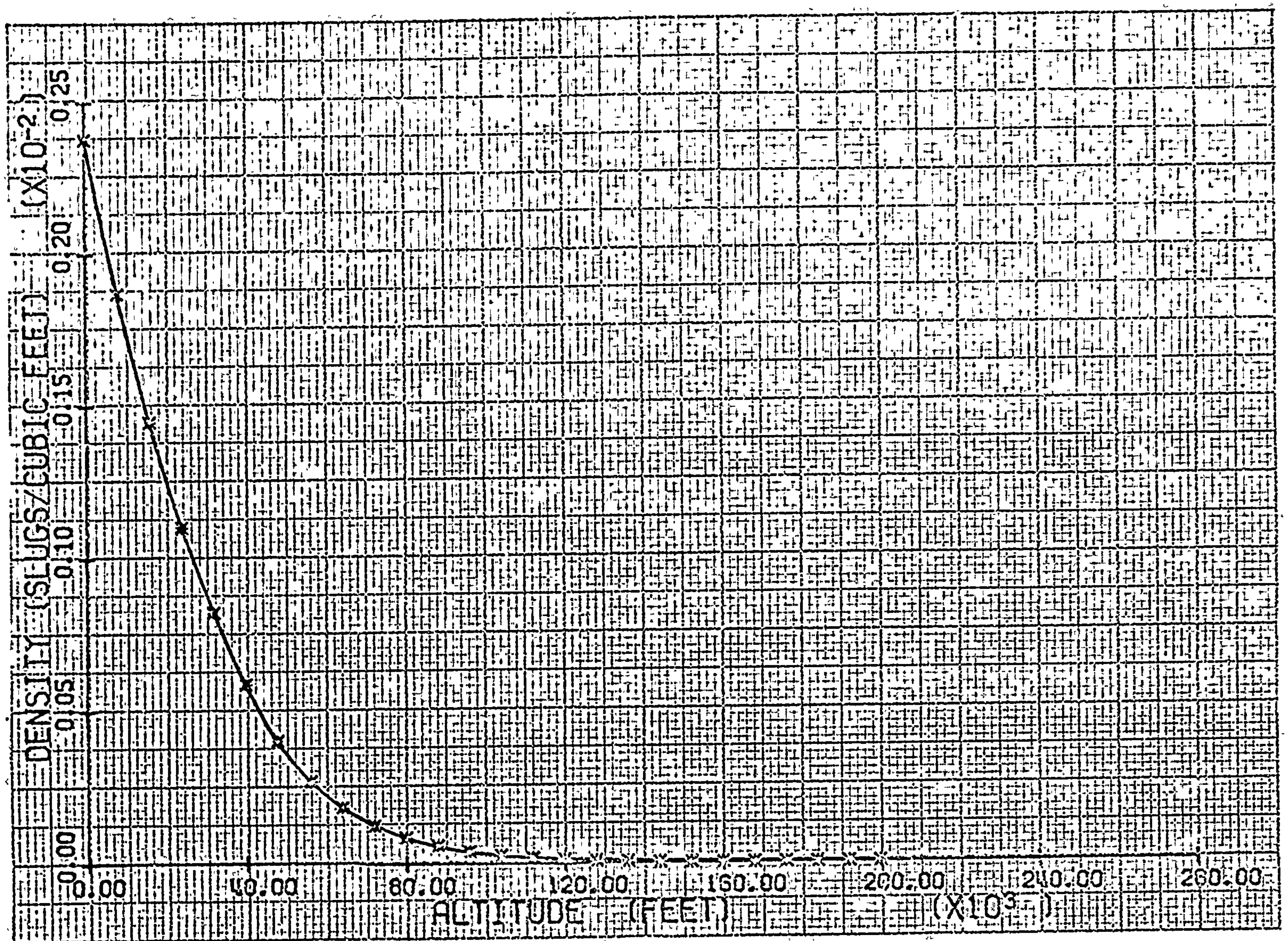


Fig. 4. Atmospheric Density

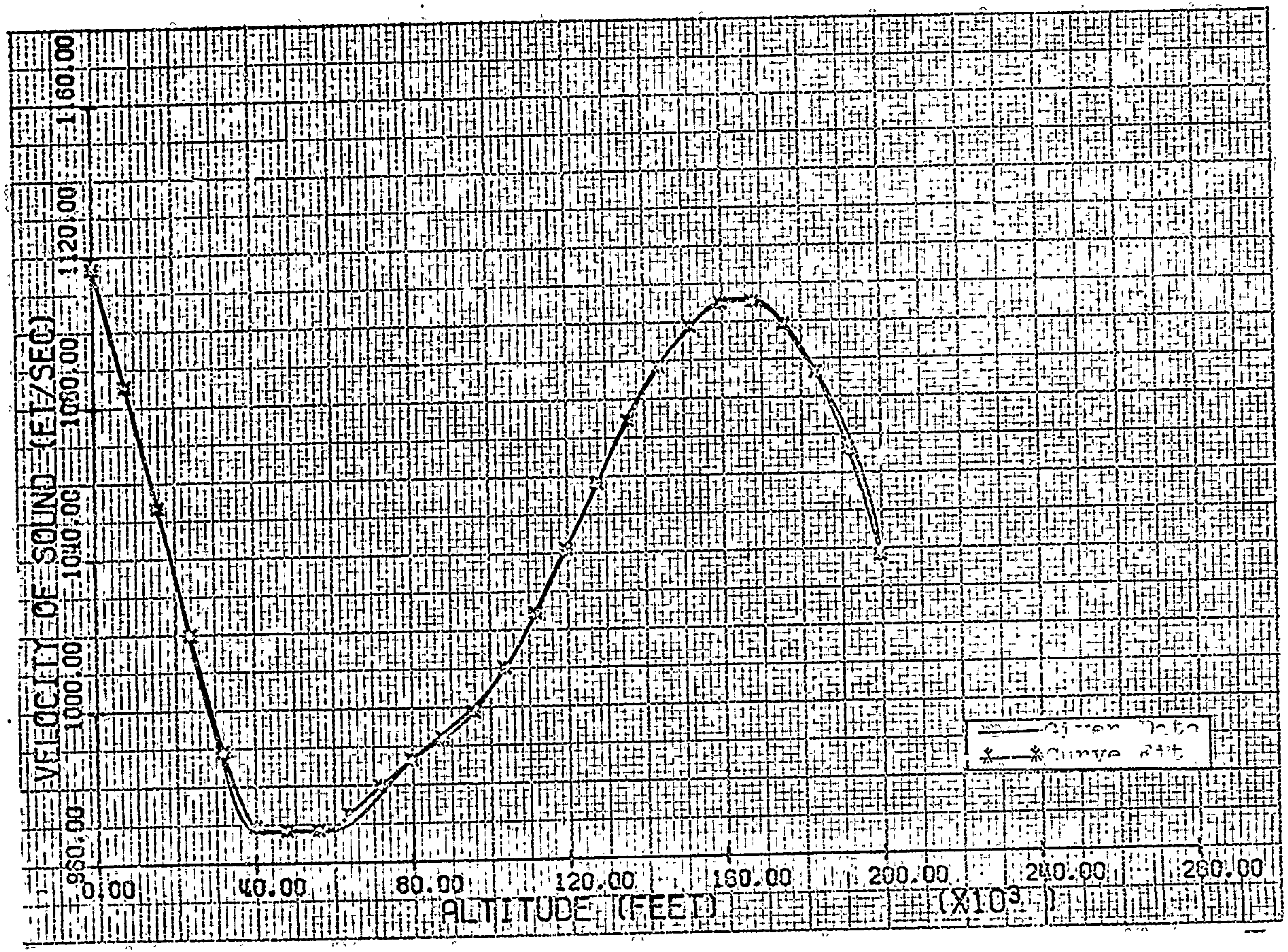


Fig. 5. Curve Fit of the Velocity of Sound

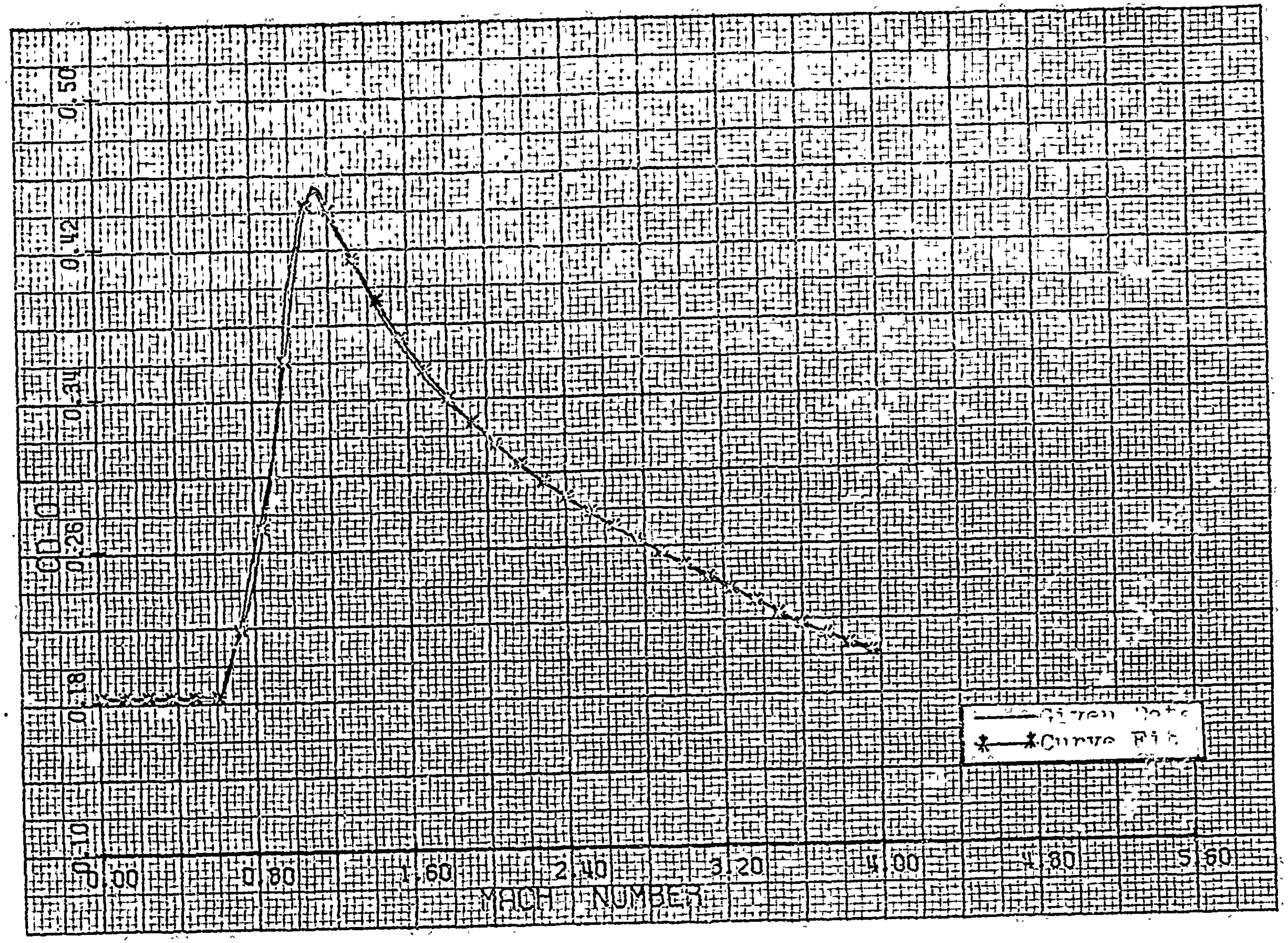


Fig. 6. Curve Fit of $C_{Do}$

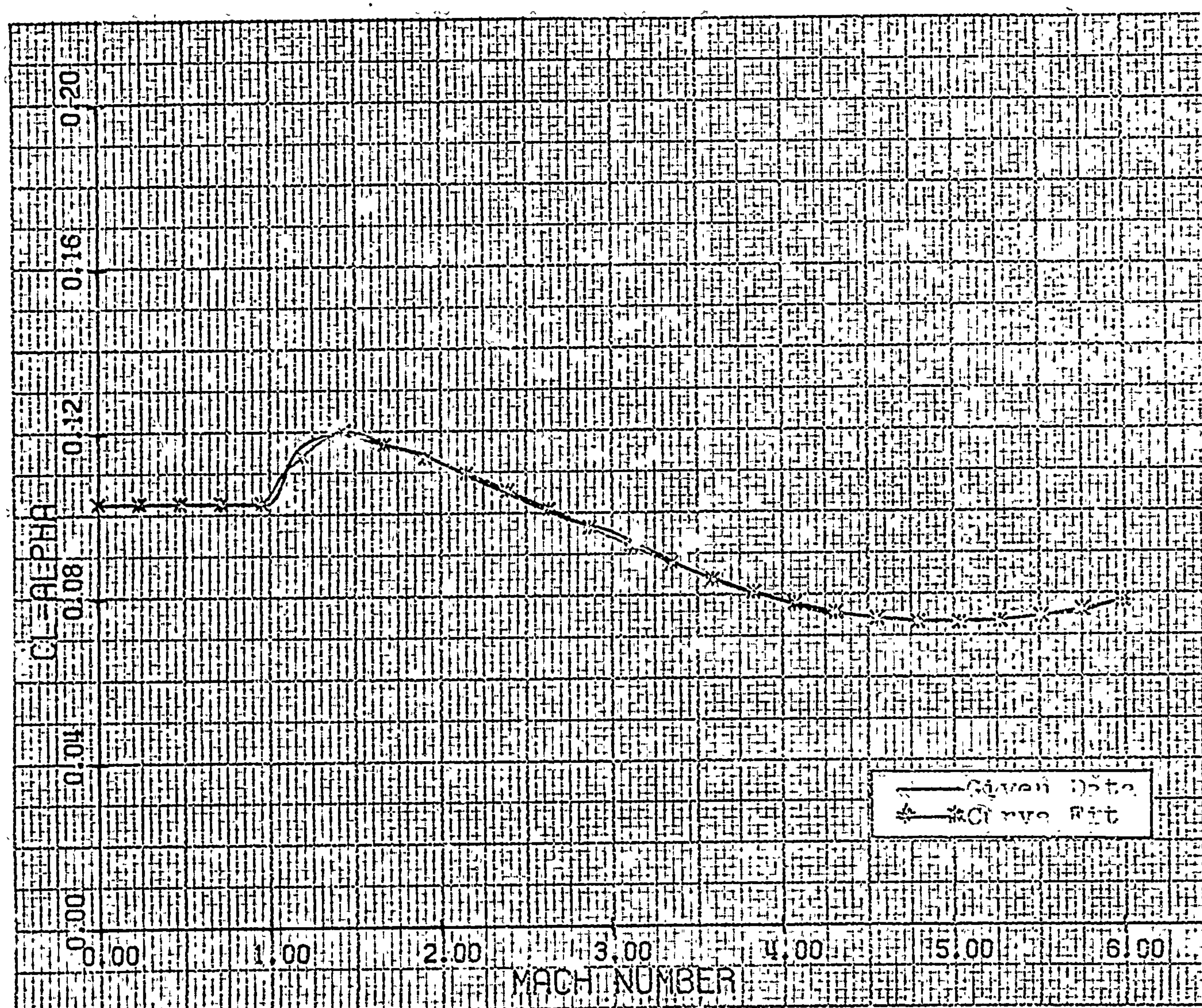


Fig. 7. Curve Fit of $C_{L\alpha}$ Per Degree

## V. The Conjugate Gradient Technique

Now that the optimal control problem has been formulated, some iterative technique must be used to solve it. In this paper the conjugate gradient method (Ref 7) is used. In general this method is able to converge quickly to a near optimal solution from poor initial guesses on control.

### The Conjugate Gradient Algorithm

(1) Choose and arbitrary $\alpha_0$
(2) Set $p_0 = Hu(\alpha_0)$ where $i = 0$
(3) Find $k_i$ such that $J(\alpha_i + k_i p_i)$ is maximized with respect to $k_i$
(4) Set $\alpha_{i+1} = \alpha_i + k_i p_i$
(5) Set $Hu_{i+1} = Hu(\alpha_{i+1})$
(6) Test the gradient for convergence; if there is convergence, stop.
(7) Set $\beta_i = \langle Hu_{i+1}, Hu_{i+1}\rangle / \langle Hu_i, Hu_i\rangle$ (inner product)
(8) Set $p_{i+1} = Hu_{i+1} - \beta_i p_i$
(9) Repeat starting with Step (3).

It is desired to find $\alpha_{i+1}$ such that $Hu_{i+1}$ is zero for all time. Because this problem is non-linear and non-quadratic, the conjugate gradient method does not find the optimal control in five iterations. Therefore, it is necessary to set some arbitrary tolerance on convergence and/or stop the program using some other condition such as the number of iterations or computer execute time. (The gradient in Step (7) is treated as an n by 1 vector where

$n = \frac{t_f}{\Delta t} + 1.)$

The Linear K-Search Algorithm

The linear K-search is the method used to perform Step (3) of the conjugate gradient algorithm. It is assumed that J is a linear function of $\alpha$ and therefore, $J' = \frac{\partial J}{\partial k_i} = p_i^T Hu(\alpha_i + k_i p_i)$. In this problem J is definitely not a linear function of $\alpha$, but it is assumed that $p_i^T Hu$ is a good approximation of $J'$.

Let $K' = \dfrac{2\{J(\alpha_i + [p_i^T p_i]^{-1} p_i) - J(\alpha_i)\}}{p_i^T Hu_i}$ and consider $p_i$ and $Hu_i$ n by 1 vectors.

(1) Initial guess of $k_i$:
(i) Let $K = K'$ if $0 < K' < (p_i^T p_i)^{-1}$
(ii) Let $K = (p_i^T p_i)^{-1}$ if $K'$ is elsewhere

(2) Evaluate $J(\alpha_i + nKp_i)$ where $(n = 0, 1, 2, 4, 8, \ldots\ a, b)$ and $p_i^T Hu(\alpha_i + bKp_i) < 0$; therefore $aK < k_i < bK$

(3) Interpolating $k_i$:

(i) Let
$$k_i = K\left\{b - \left[\frac{J'(b) + u - z}{J'(b) - J'(a) + 2}\right](b-a)\right\}$$
where
$$z = \frac{3[J(a) - J(b)]}{K(b-a)} + J'(a) + J'(b)$$
$$u = [z^2 - J'(a)\,J'(b)]^2$$
$$J(a) = J(\alpha_i + aKp_i)$$
$$J'(a) = p_i^T Hu(\alpha_i + aKp_i) = \frac{\partial J(a)}{\partial(aK)}$$

(ii) Set $\alpha_{i+1} = \alpha_i + k_i p_i$ if $J(a)$ and $J(b) < J(k_i)$

(iii) If neither $J(a)$ nor $J(b)$ is less than $J(k_i)$, return to Step 2) using $n = a + \frac{b-a}{5} j$ where $(j = 1, \ldots a', b' \ldots 5)$ and $a'K < k_i < b'K$

With the exception of Step (3-iii), this linear search is the linear search method of Fletcher and Reeves (Ref 3). The equivalence to Step (3-iii) in Fletcher and Reeves divides the interval into two subintervals at $k_i$ and tests the sign of $J(k_i)$ to determine which subinterval to use in Step (3-i). The Fletcher and Reeves method is less cumbersome, but Step (3-iii) seemed more reliable in this problem.

# VI. Results and Discussions

The maximum range obtained in this investigation is sensitive to the guesses of the fixed final time, the weighting function R, and initial control. The selection of values for these three parameters determines whether a local or global maximum range is obtained. A trial-and-error method is perhaps the only way of etermining reasonable estimates for the optimal values of these parameters.

## The Final Time

Some idea of the values for the fixed final time was obtained by computing trajectories with constant angles of attack. The values used were 0.05, 0.1, 0.15, and 0.2 radian. Judging from their impact time, it was decided to work with $t_f = 300$ seconds.

## The Weighting Function

In order to get some idea of the best magnitude of the weighting function, conjugate gradient computer runs were made with R and $\alpha_o$ equal to zero for all time, and $t_f$ equal to various values above 300 seconds. In general these runs were unstable, i.e., on many iterations control would greatly exceed 20°, the flight path angle would continuously increase, and/or the range would be negative. However, on the more stable runs the values of the gradient gave an

indication for the necessary magnitude of R. For cost $J_1$ ($J_2$) the gradient had magnitudes around $10^4$ and $10^5$ ($10^{11}$ and $10^{12}$) during thrusting, while during glide the magnitudes were around $10^3$ ($10^8$) and lower. Also, the angle of attack would exceed $20^\circ$ during thrusting, specifically during the first 10 to 15 seconds, and as expected would remain well below $20^\circ$ during glide. In order for the penalty function to have some affect on the gradient during thrust, the magnitude of R would have to be comparable to that of the gradient. The function used in this presentation was

$$R = R_0 \left[ u(t) - u(t - t_r) \right] \tag{21}$$

where $R_o$ is some number in the neighborhood of $10^4$ and $10^5$, $u(t)$ is the unit step function and $t_r$ is the approximate complete burnout time of 30.00 seconds. (In the following discussions $t_r$ will take on different values.)

<u>The Initial Guess on Control</u>

Once some idea of the magnitudes of R and $t_r$ had been obtained, the next step involved checking the sensitivity of the conjugate gradient method to various initial guesses on control. Four guesses (0.0, 0.1, 0.15, and 0.2 radian for all time) were tested by using $J_1$ and varying $R_o$ and $t_f$ until what appeared to be the best sample runs for each of the four values was obtained. On the basis of the sample runs $\alpha_o$ = 0.15 radian appeared to be the best choice.

However, each of the four guesses had one major fault: the conjugate gradient method did very little in optimizing control as time approached $t_f$, consequently, at final time large negative flight path angles were obtained. Therefore, the initial guess on control was chosen to be

$$\alpha_0 = \begin{cases} 0.15; & t \leq 315 \\ 0.001(t-315) + 0.15; & t \geq 315 \end{cases} \tag{22}$$

$\alpha_0$ was increased linearly as time approached $t_f$ so that $\gamma(t_f)$ would be increased, thus increasing range. The time of 315 seconds was chosen in Eq (22) because in that neighborhood the flight path angle reached critical negative values. It must be mentioned that many other functions for $\alpha_0$ where $t > 315$ seconds could have been used, and even the use of $t = 315$ seconds as the break point is questionable.

Results of Sample Computer Runs

Using $J_1$, Eqs (22), (21), and the intervals $1 \times 10^4 \leq R_0 \leq 1 \times 10^6$ and $300 \leq t_f \leq 380$, sample runs were made to determine smaller intervals for $R_0$ and $t_f$. If the run were unstable, then $R_0$ was increased by an amount that was somewhat proportional to the amount of instability. If the altitude at time $t_f$ were large, then $t_f$ was increased so as to decrease $h(t_f)$. Table II on the next page depicts the results of some of the more significant runs. In the table

Table II
Investigation Results

| Run No. | Type | RO | $t_r$ | $t_f$ | Iteration No. | $h(t_f)$ x $10^3$ | $(t_f)$ | $(t_f)$ x $10^2$ | $r(t_f)$ | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Sample | $4.5x10^5$ | 30 | 360 | 5 | 3.08 | -12.82 | 8.10 | 140.73 | Unstable; large gradient |
| 2 | Complete | $6x10^5$ | 30 | 360 | 23 | 5.97 | -14.34 | 9.08 | 136.12 | Slight improvement in gradient |
| 3 | Complete | $8x10^5$ | 30 | 360 | 33 | 5.33 | -13.12 | 8.85 | 136.92 | Slight improvement in gradient |
| 4 | Sample | $7x10^5$ | 30 | 365 | 7 | 2.64 | -11.21 | 8.41 | 130.78 | Slower converging than Run 3 |
| 5 | Complete | $8x10^5$ | 30 | 365 | 35 | 4.84 | -12.30 | 8.66 | 136.35 | No improvement in gradient |
| 6 | Complete | $8x10^5$ | 30 | 370 | 38 | 3.80 | -11.34 | 8.35 | 135.51 | Gradient about the same |
| 7 | Complete | $9x10^5$ | 30 | 370 | 37 | 3.96 | -11.30 | 8.32 | 136.83 | Gradient about the same |
| 8 | Complete | $8x10^5$ | 30 | 380 | 19 | 1.53 | -10.58 | 7.55 | 135.98 | No improvement in gradient |
| 9 | ½Complete | $4x10^5$ | 10 | 360 | 4 | 3.55 | -13.12 | 8.21 | 143.76 | Unstable; gradient larger than (2) through (8) |
| 10 | ½Complete | $8x10^5$ | 10 | 360 | 12 | 5.16 | -12.62 | 8.46 | 141.84 | Gradient slightly smaller than (9) |
| 11 | Sample | $4x10^5$ | 10 | 360 | 4 | | | | 121.38 | |
| 12 | Sample | $1x10^5$ | 10 | 360 | 1 | 5.52 | -12.38 | 8.42 | 143.24 | Smaller ranges as run continued |
| 13 | Sample | $2x10^5$ | 10 | 360 | 1 | 5.49 | -12.43 | 8.41 | 143.37 | |
| 14 | Sample | $4x10^5$ | 10 | 360 | 2 | 4.5 | -13.02 | 8.28 | 144.05 | Unstable, large gradient |

a sample run was one with a computer-execute-time limit of four or five minutes. These runs were used to determine if a complete run using the sampled guesses was warranted. A complete run was forty-one conjugate gradient iterations.

Using the above procedure the intervals were narrowed to $4.5 \times 10^5 \leq R_o \leq 9 \times 10^5$ and $360 \leq t_f \leq 380$. Runs (1) through (8) of Table II depict the results obtained using $R_o$ and $t_f$ within those intervals. Run (1) obtained the maximum range, however, it was unstable and the gradient of the maximum-range trajectory was very large. Runs (2) through (8) were attempts to stabilize the method and hopefully increase the maximum range obtained. In these runs stability was obtained, however, the range obtained in Run (1) was never equaled or improved and very little improvement was made on the gradient. Also, varying $t_f$ did very little to improve range maximization. In fact Run (3) with $t_f = 360$ seconds was never improved as $t_f$ was increased. Therefore, all succeeding runs were made with $t_f = 360$ seconds.

The next step involved decreasing $t_r$ over the interval $5.0 \leq t_r \leq 30$, while varying $R_o$ over the interval $1 \times 10^5 \leq R_o \leq 8 \times 10^5$. The best results were Runs (9) and (10) where the maximum range obtained was 143.76 miles. Still the gradient was large during thrust. At this point in the investigation it was discovered that the mass of the vehicle

was being treated as a function of altitude, i.e., in the computer program "g" of Eq (4) was also being used as "$g_o$" of Eq (9). Although this error affected the range of the trajectory by tenths of a mile, the value of the best $R_o$ was greatly affected. A comparison of Runs (9) and (11) gives an indication of the affects of the correction. Runs (11) through (14) used the correct value for "$g_o$".

Using a modified history of the control obtained in Run (9) as the initial guess (this control affected a range of 143.26 miles, t = 360 sec.) various runs were made in an attempt to improve the range. Runs (12) through (14) are samples of those runs. Because of the results of Runs (2) through (8), no complete runs were made and $t_r$ = 10 seconds was used. As can be seen very little improvement was made on the maximum range. Further, the gradient was not improved.

At this point it was decided to use $J_2$ and the control history obtained in Run (14) as the initial guess on control. A graph of this control history is depicted in Fig. 8. Using $t_f$ = 360 seconds, R was varied from $0.1 \times 10^{12}$ to $0.1 \times 10^{14}$ using the same scheme that was used with $J_1$. No improvement on the range was obtained. Most of the runs were unstable; however, the more stable runs were unable to search effectively in the increasing range direction. Consequently, the conjugate gradient approach was abandoned even though the optimal trajectory had not been obtained.

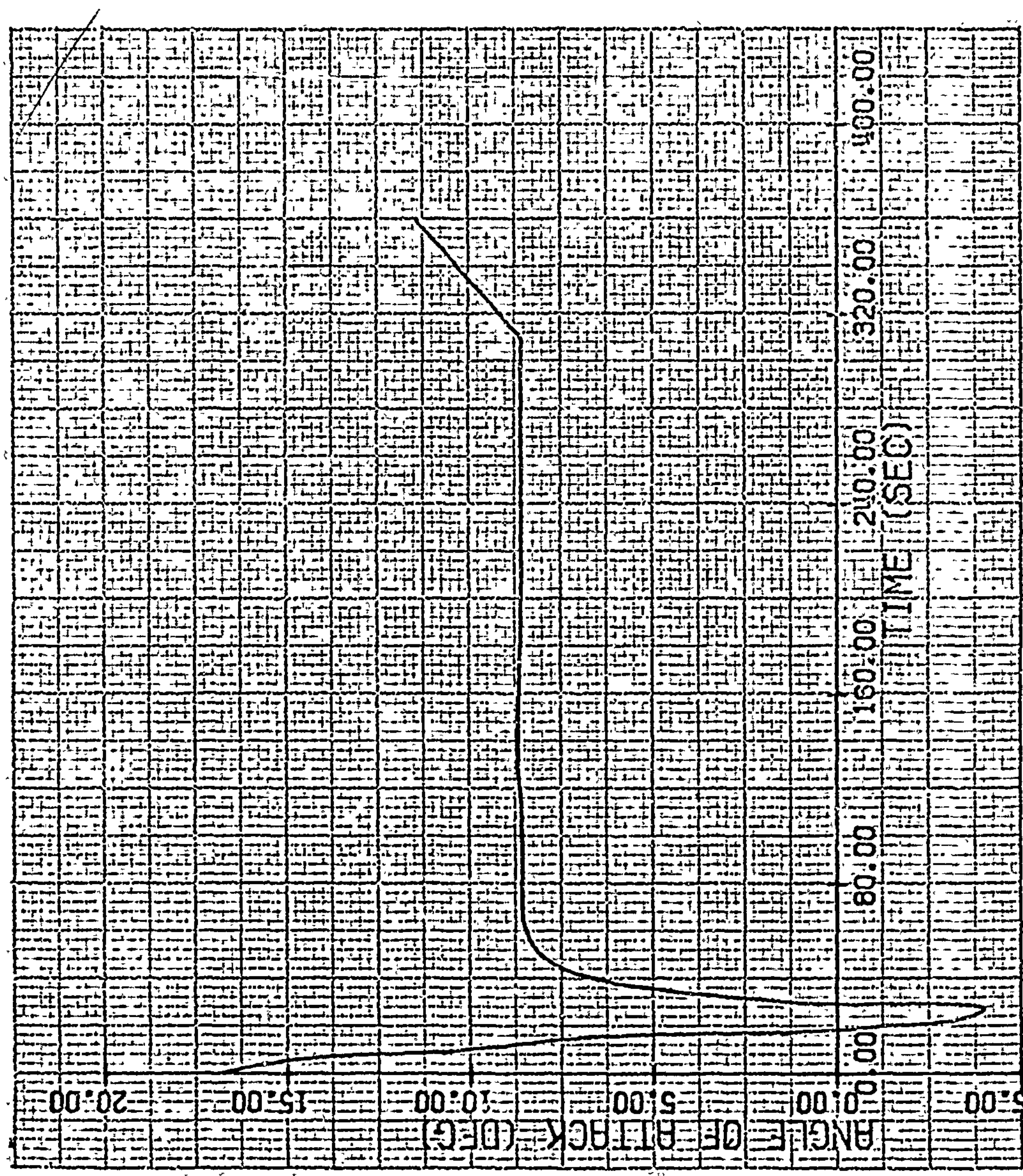


Fig. 8. Control History Run (14)

## Two Modifications

Using the control history of Run (14), two modifications were made -- (1) different constant control values were simulated for the time region $t > 65$ seconds, and (2) the conventional technique of flying the minimum drag-to-lift-ratio trajectory during glide was simulated. As a result the trajectory was improved. The control history of Run (14) obtained a total range of 147.64 miles at t = 385.73 seconds (control was held constant at 11.46° after t = 360 seconds). In modification one $\alpha =$ 0.2, 0.225, 0.25, and 0.275 radian were used. $\alpha = 0.25$ obtained the best range -- 156.79 miles at t = 450 seconds.

The necessary condition for modification two is that

$$\frac{\partial (L/D)}{\partial \alpha} = 0 \tag{23}$$

Substituting Eqs (2) and (3) into Eq (23),

$$\alpha_{LD} = \sqrt{\frac{C_{D_0}}{C_{L_\alpha}}} \tag{24}$$

where $\alpha_{LD}$ satisfies Eq (23). Using the trajectory obtained in modification one and $\alpha = \alpha_{LD}$, two trajectories were simulated -- the first used $\alpha = \alpha_{LD}$ for all of the glide, and the second used it for $t > 65$ seconds. The best range obtained was 154.27 miles where $\alpha = \alpha_{LD}$ for $t > 65$ seconds.

## The Best Trajectory Obtained

The best control history (Fig. 9.) in this investigation obtained a range of 156.79 miles. This history is

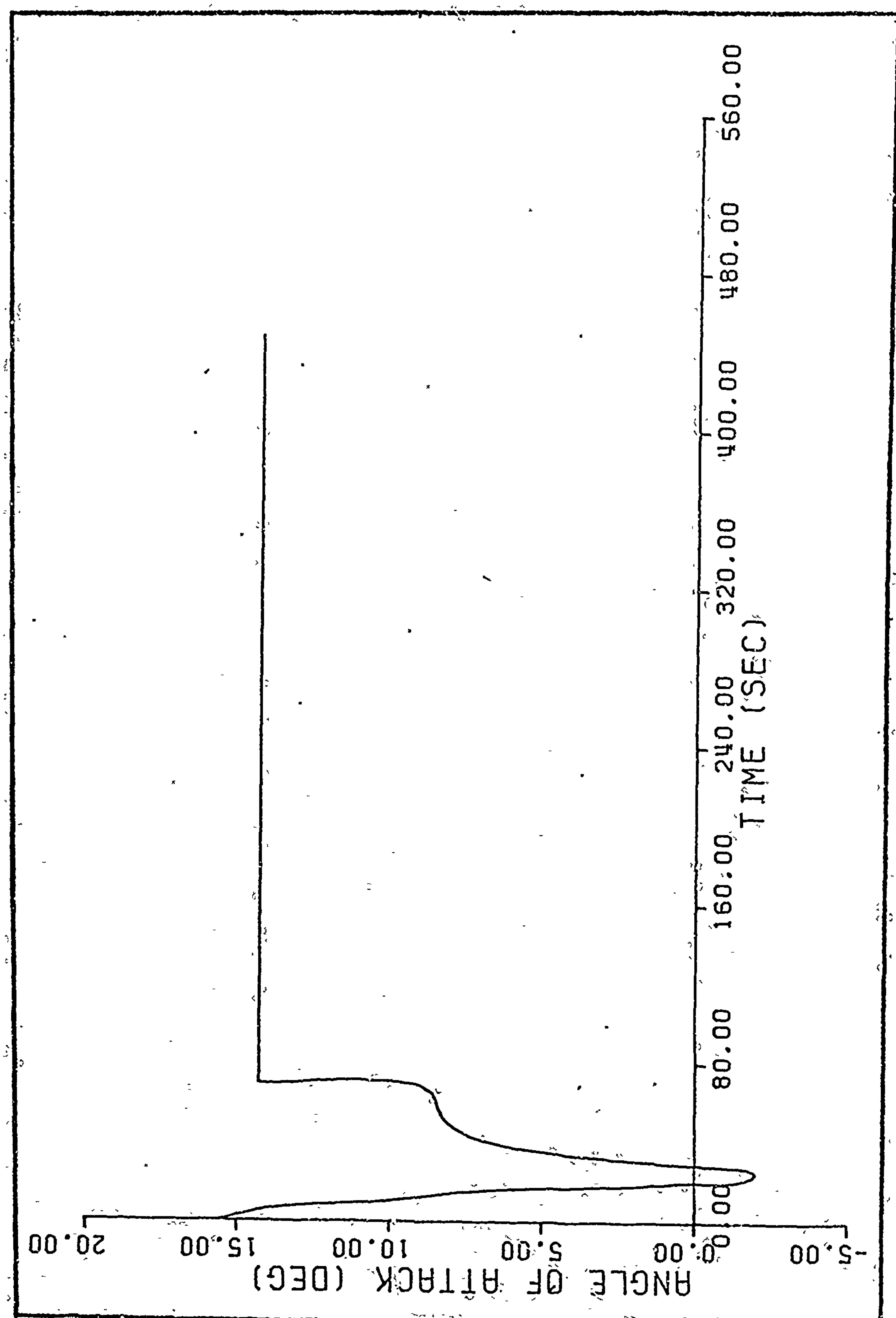


Fig. 9. Angle-of-Attack Control History

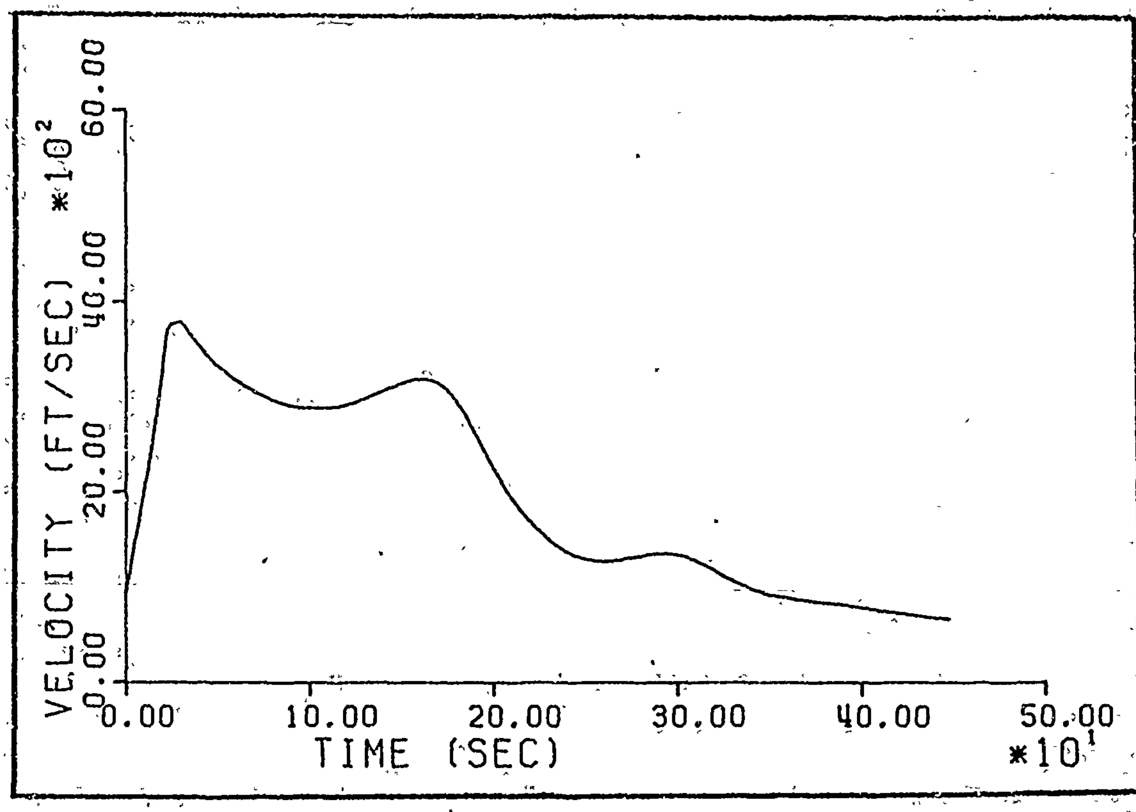


Fig. 10. Velocity vs. Time

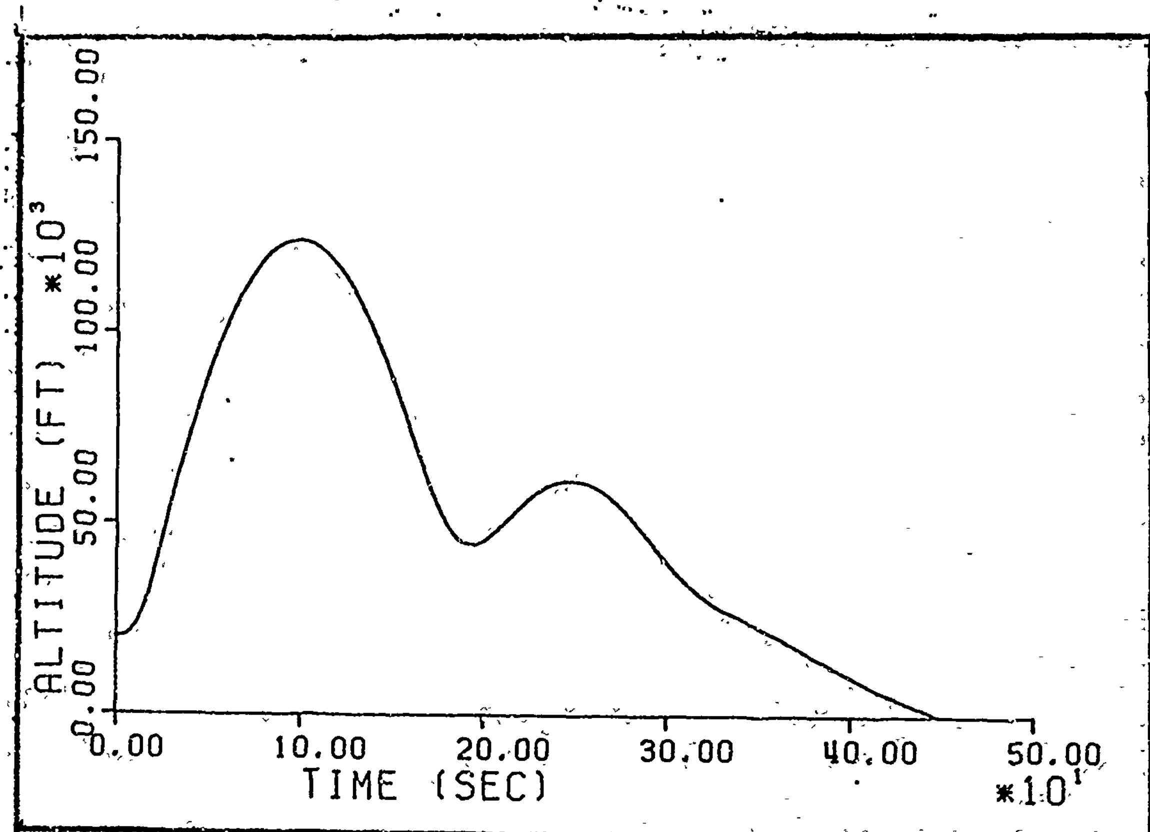


Fig. 11. Altitude vs. Time

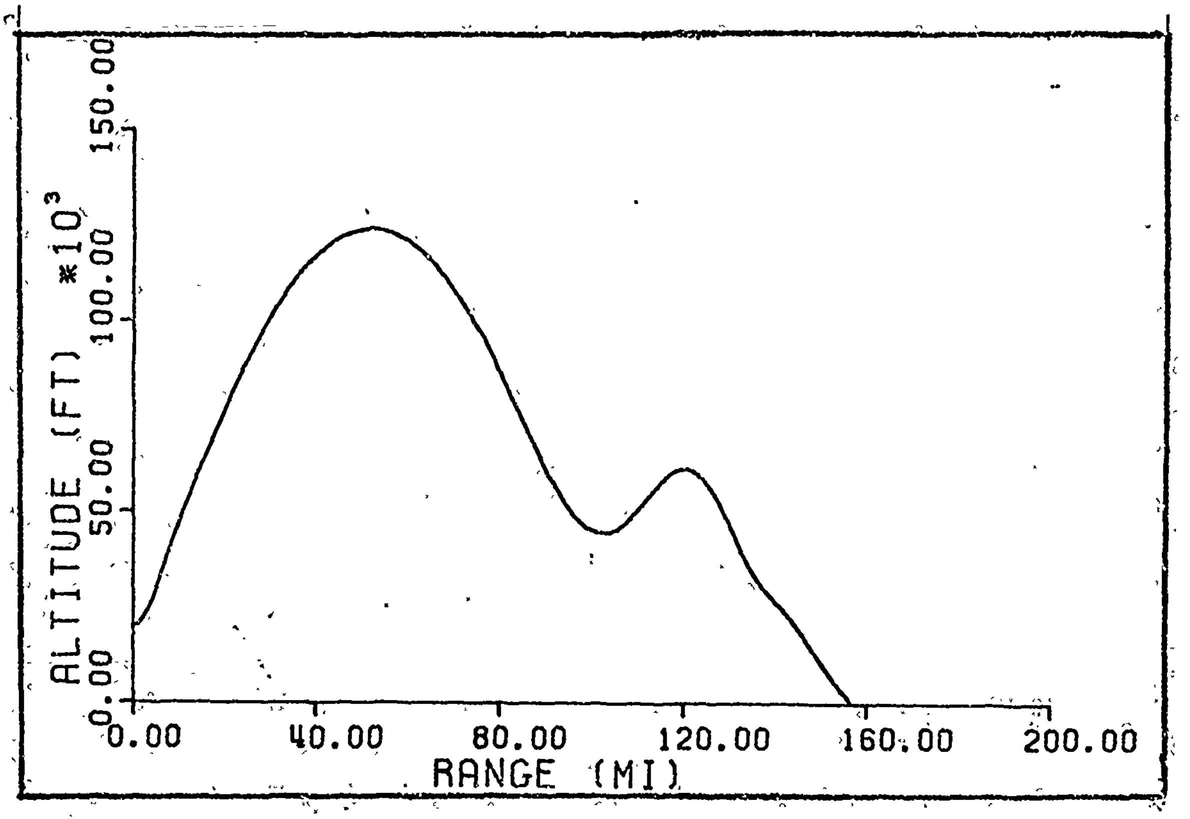


Fig. 12. Altitude vs. Range

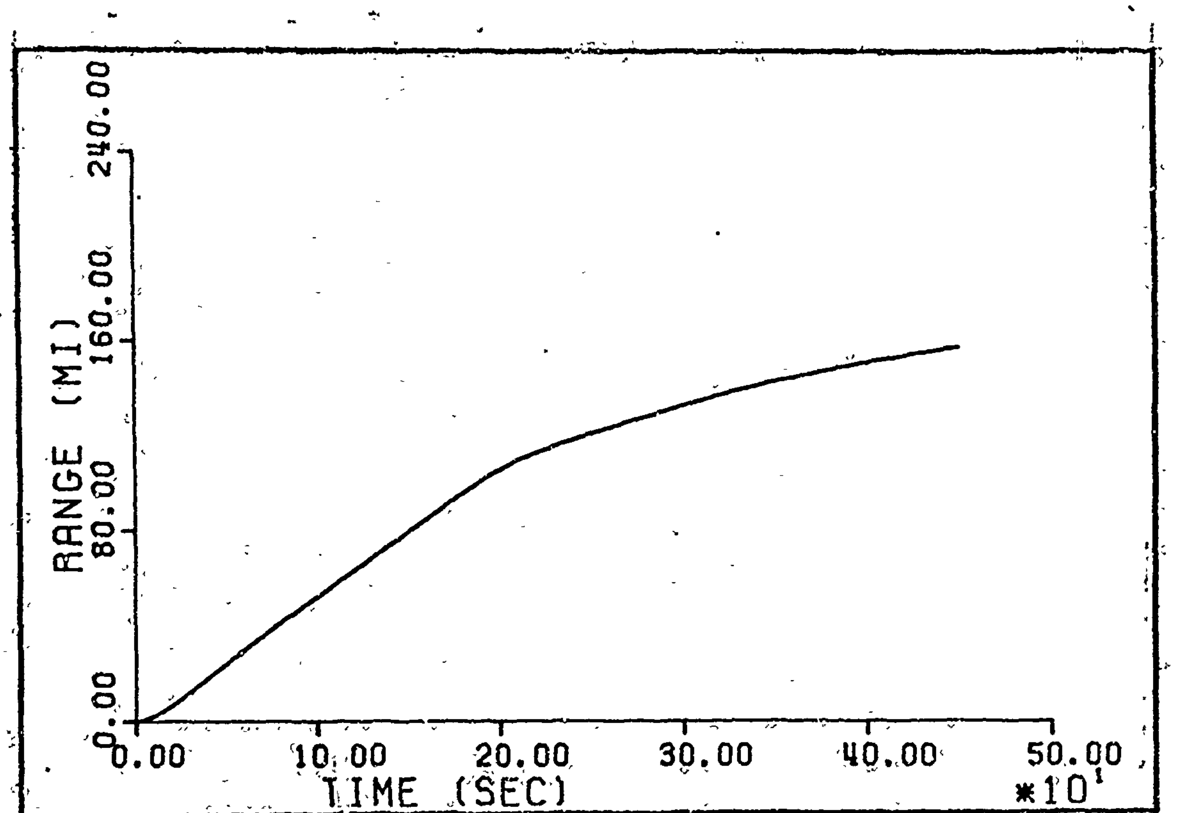


Fig. 13. Range vs. Time

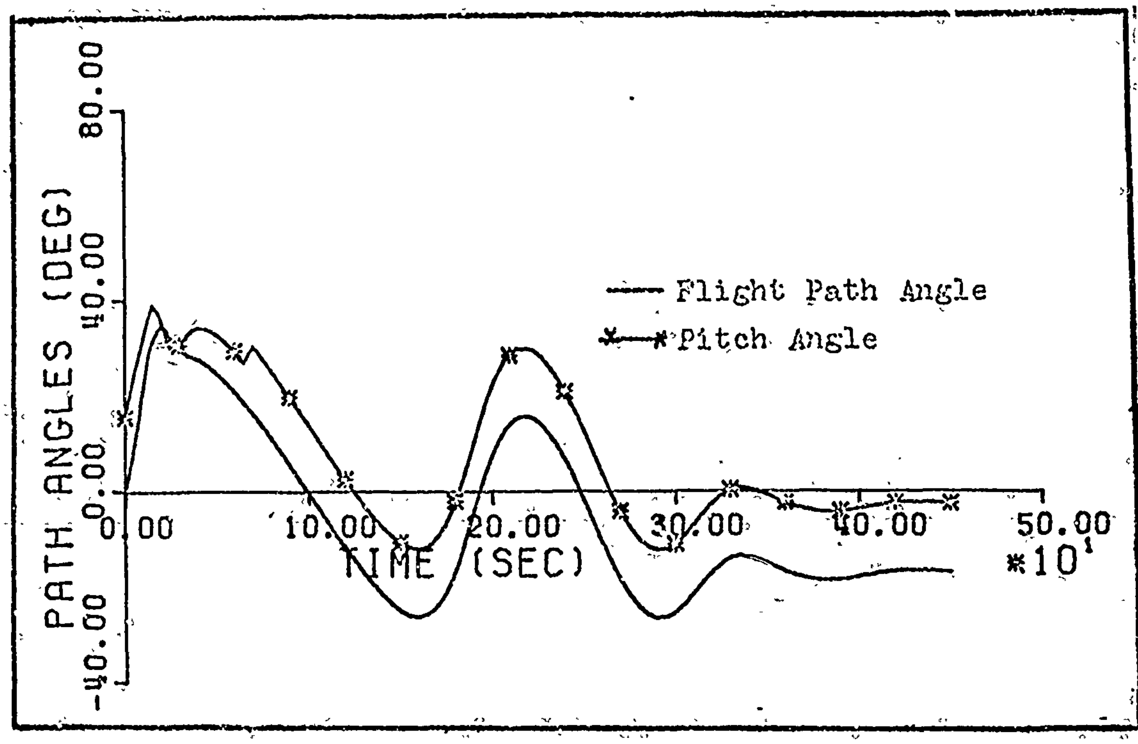


Fig. 14. Path Angles vs. Time

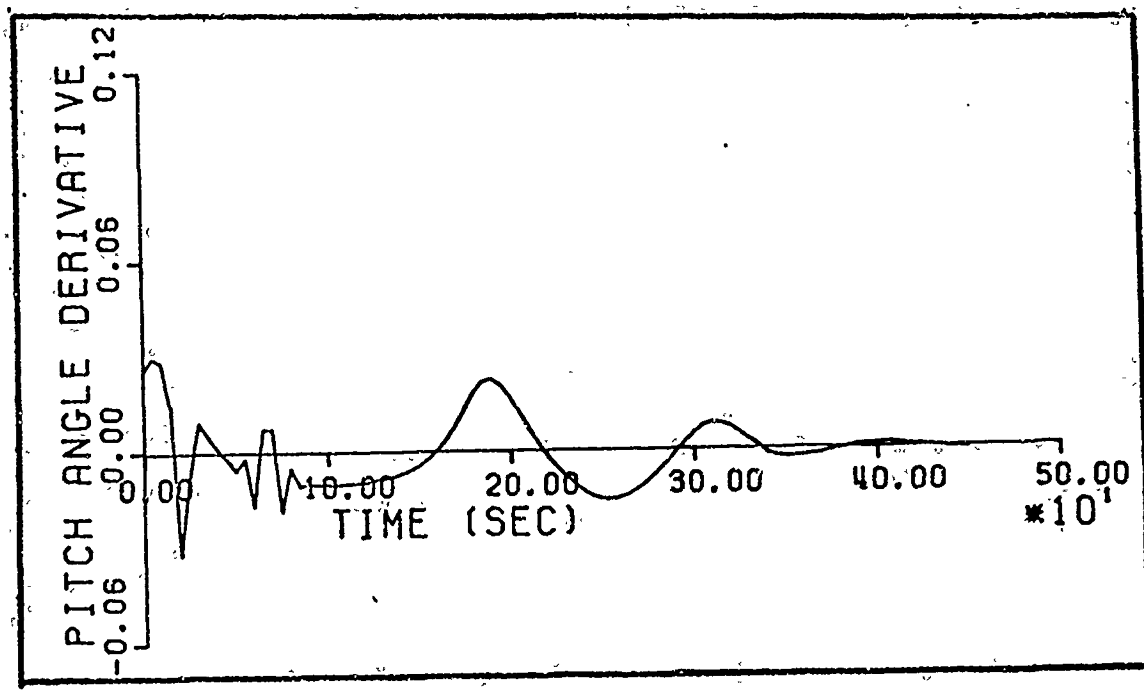


Fig. 15. Pitch Rate vs. Time

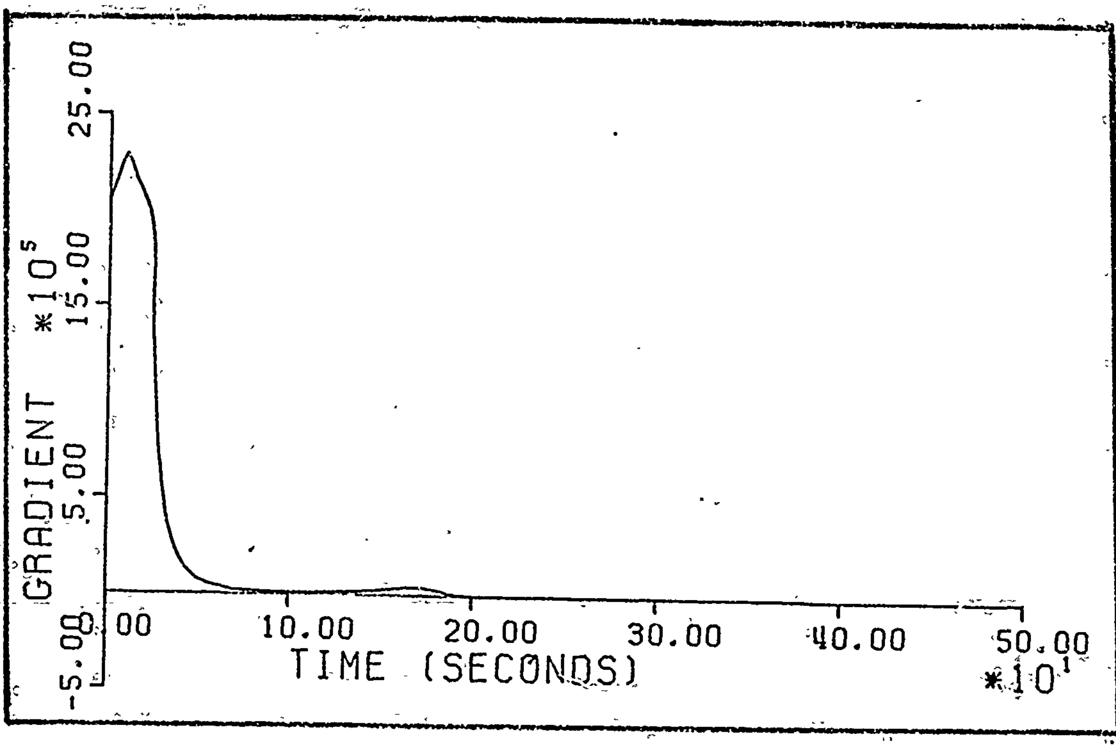


Fig. 16. Gradient vs. Time

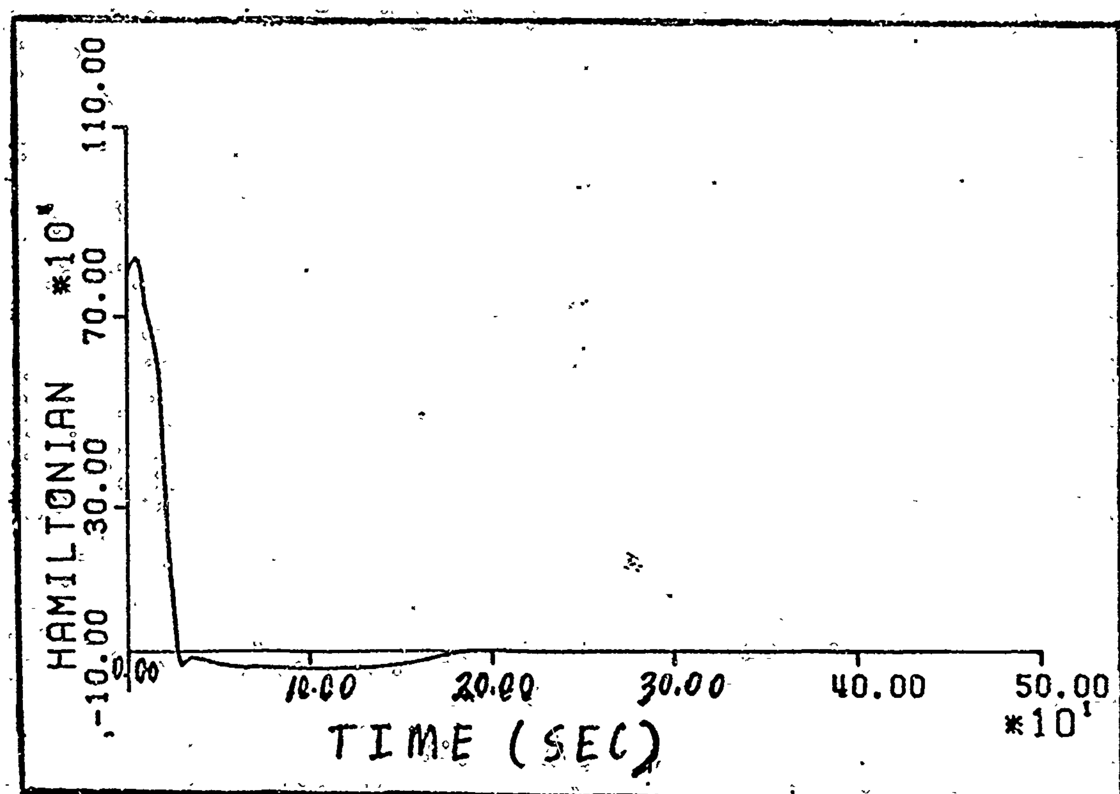


Fig. 17. The Hamiltonian vs. Time

well within the limit of $20^\circ$ since its maximum point is $16.91^\circ$. The control programs of runs in Table II that reached at least 140 miles had the same general shape. Those trajectories that fell below 140 miles had control histories with larger minimum values; consequently, higher altitudes were obtained. As expected the range is very sensitive to the angle of attack during thrust and slightly thereafter.

Figs. 10, 11, 12, 13 and 14 are graphs of the state variables. Fig. 14 includes a graph of the pitch angle $\theta$. Fig. 15 is a graph of the pitch rate. The maximum acceleration, $4.37g_o$, and the maximum pitch rate, 0.047 rad/sec$^2$ occur around 20 seconds. Both values are quite tolerable for equipment design purposes. The pitch angle at final time is equal to $-2.64^\circ$ and remains within $-3.7^\circ$ and $-2.25^\circ$ for the last 50 seconds of flight. This is quite adequate for line-of-sight and line-of-sight rate steering. The velocity at impact is Mach 0.6.

Figs. 16 and 17 are graphs of the gradient and the Hamiltonian. Both graphs indicate that the optimal trajectory has not been obtained. Note that as time approaches final time the gradient is small and the Hamiltonian is almost constant. These facts indicate the reason that the conjugate gradient method is inactive in this region. Also evident is the influence of the interval, $19.44 \leq t \leq 30.35$, where the Hamiltonian is a function of time. In these graphs $H_o = 0$.

# VII. Conclusions

A cost function including an integral penalty function to constrain control can be used to obtain an approximation of the range-maximizing, angle-of-attack control history of an air-to-surface missile trajectory using non-linear equations of motion and the conjugate gradient method.

The major shortcoming is the necessity of having to guess values for the weighting on the penalty function, the initial guess on control, and the final time. The maximum range obtained is greatly dependent upon these guesses. Another shortcoming is the inability of the method to optimize control as time approaches the guessed final time. Finally, the most obvious drawback is that the exact optimal trajectory is not obtained.

# VIII. Recommendations

Although the approach used in this investigation is one of the simplest ways of setting up the problem, obviously, other ways might lead to better results. With the exception of the last two subsections, all of the following recommendations discuss possible ways of improving the results of this paper. The subsection "Numerical Methods" treats possible ways of decreasing the computer execute time necessary for each gradient iteration, and the last subsection "The Control System" treats a possible area of further investigation.

## Objective

As with all optimal control approaches there are a number of ways of setting up the problem. As suggested by Lasdon, Mitter, and Waren (Ref 7) end conditions can be treated using penalty functions in the objective. Since it is desired that $x_3(t_f) = 0$, the objective can be written as

$$J = -R_1\left[x_3(t_f)\right]^2 + \left[x_4(t_f)\right]^2 - R\int_0^{t} \tfrac{1}{2}\alpha^2 dt \qquad (26)$$

where $R_1$ is a weighting function. Thus, the use of this objective tends to maximize the range while minimizing the altitude at time $t_f$. However, this approach is very sensitive to the values used for the final time $t_f$ and $R_1$. Of course trial-and-error methods would have to be used to determine reasonably good values for $t_f$, $R_1$, and R.

A better objective would be

$$J = R_1 \left[X_3(t_f)\right]^2 + \left[X_4(t_f)\right]^2 - R\int_0^t \tfrac{1}{2}\alpha^2\, dt \qquad (27)$$

which tends to maximize range as well as altitude at final time. This approach is less sensitive to final time.

Constraining Control

The problem of constraining the control can be handled by introducing a fifth state and dropping the penalty integral term of Eq (10). Since it is desired that $\alpha_{min} \leq \alpha \leq \alpha_{max}$, then the fifth state equation can be

$$X_5^2 = R_1 \left[(\alpha_{max} - \alpha)(\alpha - \alpha_{min})\right] \qquad (28)$$

This keeps the objective as simple as possible, but adds another state variable and adjoint state variable to the problem. Again, "brute force" techniques will have to be used to determine $R_1$.

Method of Second Variation

A more sophisticated optimal computational scheme is the second variation method (Ref 9: 414). The conjugate gradient method essentially searches for the first order effects of the control on the objective. By considering second order effects as well, the second variation method converges much more rapidly than the conjugate gradient method (Ref 7: 138). However, the second variation algorithm is much more complex and very sensitive to the initial

guess on control. (In general if the initial guess on control is not close to the optimal, the second variation method will diverge.) Of interest is the use of the angle-of-attack history obtained by this investigation as the initial guess on control in the second variation method.

## Two Control Variables

Another sophisticated approach is the use of the conjugate gradient method in solving an optimal control problem involving two control variables, specifically the angle of attack and the thrust-vector angle. (The thrust-vector angle is measured from the x -axis of the missile to the thrust vector. In this paper the thrust-vector angle is zero for all time.) During thrusting, the gradient becomes a 2 X 1 vector and the conjugate gradient k-parameter becomes a 2 X 2 matrix. After thrusting, the problem simplifies to a one-control-variable problem. Most, if not all, of the matrix subroutines needed to program this two-control-variable approach are stored on the 7044/7094 computer system library of the Digital Computation Center; WPAFB, Ohio.

## Interval Maximization

One major fault in the approach used in this investigation is the tendency for the conjugate gradient method to make little or no effort to iterate toward an optimal trajectory in the region where time approaches the final time.

Maximizing over smaller and smaller intervals of time seems to be a way of solving this problem. First maximization is attempted over the entire range of time ($t_o$, $t_f$) as was done in this paper. Then some time $t_1$ is chosen where $t_1$ represents the time beyond which the conjugate gradient method appeared to fail. Next maximization is done over the interval ($t_1$, $t_f$). Then a larger $t_1$ is chosen and maximization is repeated over that smaller time interval.

Numerical Methods

As previously mentioned, a set of Lagrangian differentiation formulas is used to calculate the approximations of the derivatives of the data. Using these formulas in the computer program is somewhat cumbersome and time consuming. A more efficient way is to use curve fits of the approximations of the derivatives. However, curve fitting the derivatives is a more delicate operation than curve fitting the given data points. (An effort to curve fit the approximations of the derivatives with polynomials resulted in curve fits that are not as accurate as those obtained in Chapter III and consequently, are not used in this presentation.) Judicious use of a combination of least square curve fits such as the polynomial and the exponential may result in curve fits with satisfactory accuracy.

In this presentation the fourth order Runge-Kutta integration formula Eq (1A) is used to integrate the entire missile trajectory. Although very accurate and quite stable,

this method is somewhat cumbersome and time consuming. Other methods, such as the Adams-Bashforth predictor-corrector method, can be just as accurate, but far less time consuming. The Runge-Kutta formula can be used to determine the first four integration points and then a faster method can be used to determine the succeeding points. Of course a more sophisticated integrating subprogram is necessary, but computer execute time is reduced.

## The Control System

Once the optimal angle-of-attack history has been obtained a further investigation treats the design of a practical control system that flys the missile along the optimal trajectory. In designing such a control system, open or closed loop, some of the aspects that must be considered are the desired accuracy, the weight of the control system, the space available on board the missile, the cost, the type of control system, the possible use of other control variables besides angle of attack, and the fact that the system will have to operate in real rather than standard atmospheric conditions. Finally, one decision that must be made is whether to use classical, optimal, or stochastic control design techniques. All three have their own advantages and disadvantages.

## Bibliography


1. Denham, Walter F. "Range Maximization of a Surface-to-Surface Missile with In-Flight Inequality Constraints." Journal of Spacecraft and Rockets, 1, No. 1: 78 - 83 (January 1964).

2. Dommasch, D.O., S.S. Sherby and T.F. Connolly. "The Atmosphere" in Airplane Aerodynamics. New York: Pitman Publishing Corporation, 1951.

3. Fletcher, R. and C.M. Reeves. "Function Minimization By Conjugate Gradients." The British Computer Journal: 149 - 154 (July, 1964).

4. Hart, William L. "Advanced Partial Differentiation" in Calculus. Boston: D.C. Heath and Co., 1955.

5. Kelly, Louis G. "Curve Fitting and Data Smoothing" in Handbook of Numerical Methods and Applications. Reading, Mass.: Addison-Wesley Publishing Co., 1967.

6. - - - - "Differentiation and Integration" in Handbook of Numerical Methods and Applications. Reading, Mass.: Addison-Wesley Publishing Co., 1967.

7. Lasdon, L.S., S.K. Mitter, and A.D. Waren. "The Conjugate Gradient Method for Optimal Control Problems." IEEE Transactions on Automatic Control, AC-12, No.2.: 132 - 138 (April, 1967).

8. Morrison, Richard B. and Melva J. Ingle (Ed.). "Atmospheric Properties" in Design Data for Aeronautics and Astronautics. New York: John Wiley And Sons, Inc., 1962.

9. Sage, Andrew P. "Direct Computational Methods in Optimum Systems Control" in Optimum Systems Control. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1968.

10. Warfare Systems School. Spare Handbook. Maxwell Air Force Base, Alabama: Air University, July 1967.

# Appendix A

## Computer Program

The following computer program, written in Fortran IV, takes on the average less than 0.5 minutes of execute time ($t_f = 360$; $t = 5$) to perform a conjugate gradient iteration on the IBM 7044/7094 II Direct Coupled Operating System of the Digital Computation Division; Aeronautical Systems Division, Wright-Patterson AFB, Ohio. The program is composed of twelve subprograms:

(1) MAIN - performs the conjugate gradient algorithm

(2) NEWCON - performs the k-search, consequently finding the new guess on control

(3) GRADNT - computes the gradient

(4) EQUAT - uses the Lagrangian formulas to differentiate the aerodynamic data; contains the state and adjoint state differential equations

(5) FMAS - contains the equations for the mass of the missile

(6) FTHRUS - contains the equations of thrust

(7) FRHO - contains the equations approximating atmospheric density

(8) FVS - contains the polynomials approximating the velocity of sound

(9) FCLA - contains the polynomials approximating the $C_{L\alpha}$ curve of the missile

(10) FCDO - contains the polynomials approximating the $C_{Do}$ curve of the missile

(11) INTEG - uses Runge-Kutta fourth order formula to integrate the state and adjoint

state equations; calculates the objective

(12) PRICE - uses the expanded Simpson integration formula to integrate the penalty function

The Runge-Kutta formula used is

$$\bar{X}_{n+1} = \bar{X}_n + \frac{1}{6}(\bar{K}_1 + 2\bar{K}_2 + 2\bar{K}_3 + \bar{K}_4) \tag{1A}$$

where

$$\bar{K}_1 = \Delta t\left[\dot{\bar{X}}(t_n, X_n)\right]$$

$$\bar{K}_2 = \Delta t\left[\dot{\bar{X}}\left(t_n + \frac{\Delta t}{2}, \bar{X}_n + \frac{\bar{K}_1}{2}\right)\right]$$

$$\bar{K}_3 = \Delta t\left[\dot{\bar{X}}\left(t_n + \frac{\Delta t}{2}, \bar{X}_n + \frac{\bar{K}_2}{2}\right)\right]$$

$$\bar{K}_4 = \Delta t\left[\dot{\bar{X}}(t_n + \Delta t, \bar{X}_n + \bar{K}_3)\right]$$

The Simpson integration formula is

$$\int_{t_0}^{t_n} f(t)\,dt = \frac{\Delta t}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots + 4f_{n-1} + f_n) \tag{2A}$$

where

$$f_n = f(t_n)$$

$$t_{n+1} = t_n + \Delta t$$

```
$IBJOB          MAP
$IBFTC MAIN
      COMMON/TOO/ DTIME,NSTEP,NSTATE,SAREA,COST,FTIM,FCONTR,XVOST,RO,TI
      DIMENSION X(8,100),CONTR(100),TIME(100),DHDU(100),DHDU1(100),
     1S(100),A(100),B(100)
      READ(5,100) NSTEP,NSTATE,ITER,DTIME,TOL,TI,RO
  100 FORMAT(3I5,3F10.5,F10.2)
      IJK=0
      SAREA=2.18
      NSTAT2=NSTATE*2
      NSTAT1=NSTATE+1
      NSTEP3=NSTEP+3
      READ(5,101) (X(I,1),I=1,NSTATE),(X(I,NSTEP3),I=NSTAT1,NSTAT2)
  101 FORMAT(4E15.8)
      TIME(1)=0.0
      NSTEP1=NSTEP+1
      DO 11 I=2,NSTEP1
   11 TIME(I)=TIME(I-1) + DTIME
      CONTR(1)=15.61/57.2958
      CONTR(2)=14.37/57.2958
      CONTR(3)=10.37/57.2958
      CONTR(4)=7.57 /57.2958
      CONTR(5)= 0.77/57.2958
      CONTR(6)=-1.99/57.2958
      CONTR(7)= 1.05/57.2958
      CONTR(8)= 4.65/57.2958
      CONTR(9)= 6.63/57.2958
      CONTR(10)=7.62/57.2958
      CONTR(11)=8.10/57.2958
      CONTR(12)=8.35/57.2958
      CONTR(13)=8.48/57.2958
      CONTR(14)=8.61/57.2958
      DO 30I=15,100
   30 CONTR(I)=0.25
      CALL INTEG(TIME,X,CONTR)
      CALL GRADNT(TIME,X,CONTR,DHDU)
      COST=2.0*(COST+XVOST)
      COST=SQRT(COST)/5280.
      XVOST=XVOST/5280.**2
      WRITE(6,107)  COST,XVOST
  107 FORMAT(14H INITIAL COST=,1PE15.8,15H INITIAL XVOST=,1PE15.8)
      XVOST=XVOST*5280.**2
      COST=COST*5280.
      COST=0.5*COST**2-XVOST
      DO 1 I=1,NSTEP1
      DHDU1(I)=DHDU(I)
    1 S(I)=DHDU(I)
    9 CALL NEWCON(TIME,X,CONTR,S,DHDU)
      IJK=IJK+1
      WRITE(6,102) IJK
  102 FORMAT(15H1ITERATION NO.,I5//90H   TIME        VELOCITY       PATH AN
     1G     ALTITUDE          RANGE       CONTROL              GRADIENT/69H    (S
     2EC)         (FT/SEC)          (DEG)           (FT)             (MI)          (DEG)/)
      DO 2 I=1,NSTEP1
      X(2,I)=X(2,I)*57.2958
      X(4,I)=X(4,I)/5280.
      CONTR(I)=CONTR(I)*57.2958
```

```
      WRITE(6,103) TIME(I),X(1,I),X(2,I),X(3,I),X(4,I),CONTR(I),DHDU(I)
103   FORMAT(F8.2,4X,F9.2,4X,F7.2,4X,F10.2,4X,F8.2,4X,F7.2,10X,1PE15.8)
      X(2,I)=X(2,I)/57.2958
      X(4,I)=X(4,I)*5280.
      CONTR(I)=CONTR(I)/57.2958
      IF(TIME(I).GT.FTIM) GO TO 10
2     CONTINUE
10    COST=COST/5280.**2
      WRITE(6,108) COST
108   FORMAT(//11H TRUE COST=, F8.2)
      COST=COST*5280.**2
      FCONTR=FCONTR*57.2958
      RANGE=2.0*(COST+XVOST)
      RANGE=SQRT(RANGE)/5280.
      WRITE(6,104) RANGE, FTIM,FCONTR
104   FORMAT(7H RANGE=, F8.2,    12H FINAL TIME=,F8.2,32H VALUE OF CONTRO
     1L AT FINAL TIME=,F8.2)
      FCONTR=FCONTR/57.2958
      XVOST=XVOST/5280.**2
      WRITE(6,106) XVOST
106   FORMAT(7H XVOST=,1PE15.8)
      XVOST=XVOST*5280.**2
C
C  TEST  FOR  CONVERGENCE
C
12    TGRAD=0.0
      DO 3 I=1,NSTEP1
      DH=ABS(DHDU(I))
3     IF(DH.GT.TGRAD) TGRAD=DH
      WRITE(6,105)  TGRAD,TOL
105   FORMAT(44H MAX. ABSOLUTE  VALUE  OF  THE  GRADIENT  IS,1PE15.8/
     119H THE  TOLERANCE  IS,1PE15.8)
      IF(IJK.GT.ITER)  GO TO 4
      IF(TOL.GE.TGRAD)  GO TO 4
C
C   FINDING  NEW  SEARCH  DIRECTION
C
      DH=0.
      DH1=0.
      DO 5 I=1,NSTEP1
      DH=DHDU(I)**2+DH
5     DH1=DHDU1(I)**2+DH1
      BETA=DH/DH1
      DO 8 I=1,NSTEP1
      S(I)= DHDU(I)+BETA*S(I)
8     DHDU1(I)=DHDU(I)
      GO TO 9
4     STOP
      END
      SUBROUTINE NEWCON(TIME,X,CONTR,S,DHDU)
      COMMON/TOO/ DTIME,NSTEP,NSTATE,SAREA,COST,FTIM,FCONTR,XVOST,RO,TI
      DIMENSION X(8,100),CONTR(100),TIME(100),DHDU(100),S(100)
      DO 131 I=1,NSTEP1,4
131   WRITE(6,130) TIME(I),S(I)
130   FORMAT(F10.2,5X,1PE15.8)
C
C   ESTIMATE  OF  ORDER  OF  MAGNITUDE  OF  K
```

```
C
      TCOST = COST
      XK=0.0
      NSTEP1=NSTEP+1
      DTCOST=0.0
      DO 1 I=1,NSTEP1
      DTCOST=DTCOST+S(I)*DHDU(I)
1     XK=XK+S(I)**2
      XK=1.0/SQRT(XK)
      DO 2 I=1,NSTEP1
2     CONTR(I)=CONTR(I) + XK*S(I)
      CALL INTEG(TIME,X,CONTR)
      WRITE(6,121) TCOST,COST
121   FORMAT(//7H TCOST=,1PE15.8 ,4X,6H COST=,1PE15.8//)
      H=2.0*(COST-TCOST)/DTCOST
      IF(0.0.GE.H.OR.H.GE.XK)  H=XK
C
C   FINDING INTERVAL WHERE THE DERIVATIVE OF THE COST PASSES
C   THROUGH ZERO
C
      DO 4 I=1,NSTEP1
4     CONTR(I)=CONTR(I)-XK*S(I)
      DO 6 I=1,20
      DO 20 K=1,NSTEP1
20    CONTR(K)=CONTR(K)+2.0**(I-1)*H*S(K)
      WRITE(6,105)  I
105   FORMAT(22H TWO IS RAISED TO THE ,I3,9H -1 POWER)
      CALL INTEG(TIME,X,CONTR)
      CALL GRAPNT(TIME,X,CONTR,DHDU)
      DCOST=0.0
      DO 5 K=1,NSTEP1
5     DCOST=DCOST+S(K)*DHDU(K)
      WRITE(6,107) COST,TCOST
107   FORMAT(/6H COST=,1PE15.8,7H TCOST=,1PE15.8/)
      DO 21 K=1,NSTEP1
21    CONTR(K)=CONTR(K)-2.0**(I-1)*H*S(K)
      IF(DCOST.LT.0.0)  GO TO 7
      IF(I.EQ.20) WRITE(6,100)
100   FORMAT(18H TROUBLE IN SEARCH)
      IF(I.EQ.20) STOP
      TCOST=COST
6     DTCOST=DCOST
C
C   CUBIC INTERPOLATION OF K
C
7     A=2.0**(I-1)*H
      B=0.0
      IF(I.NE.1)  B=2.0**(I-2)*H
      NCI=0
9     Z=DCOST+DTCOST+3.0*(COST-TCOST)/(B-A)
      W=Z**2-DCOST*DTCOST
      W=SQRT(W)
      XK=B-(B-A)*(DTCOST+W-Z)/(DTCOST-DCOST+2.0*W)
      WRITE(6,101) XK,A,B
101   FORMAT(4H XK=,1PE15.8,3X,3H A=,1PE15.8,3X,3H B=,1PE15.8//)
      DO 10 I=1,NSTEP1
10    CONTR(I)=CONTR(I)+XK*S(I)
```

```
      IF(NCI.EQ.1) GO TO 13
      STORE=COST
      CALL INTEG(TIME,X,CONTR)
      COSTXK=COST
      COST=STORE
      CALL GRADNT(TIME,X,CONTR,DHDU)
      IF(COST.LT.COSTXK.AND.TCOST.LT.COSTXK)  GO TO 12
      TDEL=P
      DO 22 I=1,NSTEP1
22    CONTR(I)=CONTR(I)-XK*S(I)
      DO 25 I=1,5
      FAC=I
      DEL=B+FAC*(A-B)/5.0
      DO 26 K=1,NSTEP1
26    CONTR(K)=CONTR(K)+DEL*S(K)
      CALL INTEG(TIME,X,CONTR)
      CALL GRADNT(TIME,X,CONTR,DHDU)
      DCOST=0.0
      DO 28 K=1,NSTEP1
28    DCOST=DCOST+DHDU(K)*S(K)
      DO 29 K=1,NSTEP1
29    CONTR(K)=CONTR(K)-DEL*S(K)
      IF(DCOST.LT.0.0.OR.I.EQ.5)  GO TO 30
      TDEL=DEL
      DTCOST=DCOST
25    TCOST=COST
30    A=DEL
      B=TDEL
      NCI=1
      GO TO 9
12    COST=COSTXK
      RETURN
13    CALL INTEG(TIME,X,CONTR)
      CALL GRADNT(TIME,X,CONTR,DHDU)
      RETURN
      END
$IBFTC GR
      SUBROUTINE GRADNT(TIME,X,CONTR,DHDU)
      COMMON/TOO/ DTIME,NSTEP,NSTATE,SAREA,COST,FTIM,FCONTR,XVOST,RO,TI
      DIMENSION X(8,100),CONTR(100),TIME(100),DHDU(100)
      NSTEP1=NSTEP+1
      DO 1 I=1,NSTEP1
      TIM=TIME(I)
      IF(TIM.GT.TI)  R=0.0
      IF(TIM.LE.TI)  R=RO
      X3=X(3,I)
      GRAVTY=1.41002019E16/(20.9E6+X3)**2
      XMASS=FMASS(TIM)
      THRUST=FTHRUS(TIM)
      RHO=FRHO(X3)
      VS=FVS(X3)
      XM=X(1,I)/VS
      CLA=FCLA(XM)
      Q=SAREA*RHO*X(1,I)**2/2.0
C
C     THE GRADIENT EQUATION
C
```

```
1     DHDU(I)=-X(5,I)*(THRUST*SIN(CONTR(I))+CLA*CONTR(I)**2.0*Q)/XMASS
     1+X(6,I)*(Q*CLA+THRUST*COS(CONTR(I)))/(X(1,I)*XMASS)
     2-R*CONTR(I)
      DO 2 I=1,NSTEP1,4
2     WRITE(6,100)  TIME(I),DHDU(I)
100   FORMAT(F10.2,5X,1PE15.8)
      RETURN
      END
      SUBROUTINE EQUAT(TX,DXDT,TIM,U)
      COMMON/TOO/ DTIME,NSTEP,NSTATE,SAREA,COST,FTIM,FCONTR,XVOST(20),TI
      DIMENSION TX(20),DXDT(20),Y(10)
      X1=TX(1)
      X2=TX(2)
      X3=TX(3)
      X4=TX(4)
      X5=TX(5)
      X6=TX(6)
      X7=TX(7)
      X8=TX(8)
      GRAVTY=1.41002019E16/(20.9E6+X3)**2
      THRUST=FTHRUS(TIM)
      XMASS=FMASS(TIM)
      RHO=FRHO(X3)
      VS=FVS(X3)
      XM=X1/VS
      CLA=FCLA(XM)
      CDO=FCDO(XM)
      Q=SAREA*RHO*X1**2/2.0
      DRAG=(CDO+CLA*U**2)*Q
      XLIFT=CLA*U*Q
C
C     FINDING  THE  DERIVATIVES  OF  RHO,  VS,  CLA,  AND  CDO
C
      DO 2 K=1,4
      IF(K.EQ.2.AND.4.E4.LT.X3.AND.X3.LT.6.E4)  DVSDH=0.0
      IF(K.EQ.2.AND.4.E4.LT.X3.AND.X3.LT.6.E4)  GO TO 2
      IF(K.EQ.2.AND.1.6E5.LT.X3.AND.X3.LT.1.7E5)  DVSDH=0.0
      IF(K.EQ.2.AND.1.6E5.LT.X3.AND.X3.LT.1.7E5)   GO TO 2
      IF(K.EQ.3.AND.XM.LT.1.0)  DCLADM=0.0
      IF(K.EQ.3.AND.XM.LT.1.0)   GO TO 2
      IF(K.EQ.4.AND.XM.LT.0.6)  DCDODM=0.0
      IF(K.EQ.4.AND.XM.LT.0.6) GO TO 2
      IF(K.LE.2) DELTAX=750.
      IF(K.GE.3) DELTAX=0.06
      DO 1 I=1,9
      FACT=I
      IF(K.LE.2)  XT=X3+DELTAX*(FACT-5.0)
      IF(K.GE.3)  XT=XM+DELTAX*(FACT-5.0)
      IF(K.EQ.1)  Y(I)=FRHO(XT)
      IF(K.EQ.2)  Y(I)=FVS(XT)
      IF(K.EQ.3)  Y(I)=FCLA(XT)
1     IF(K.EQ.4)  Y(I)=FCDO(XT)
      DYDX=(-25.*Y(5)+48.*Y(6)-36.*Y(7)+16.*Y(8)-3.*Y(9))/(12.*DELTAX)
      DYDX=DYDX+(-3.*Y(4)-10.*Y(5)+18.*Y(6)-6.*Y(7)+Y(8))/(12.*DELTAX)
      DYDX=DYDX+(Y(3)-8.*Y(4)+8.*Y(6)-Y(7))/(12.*DELTAX)
      DYDX=DYDX+(-Y(2)+6.*Y(3)-18.*Y(4)+10.*Y(5)+3.*Y(6))/(12.*DELTAX)
      DYDX=DYDX+(3.*Y(1)-16.*Y(2)+36.*Y(3)-48.*Y(4)+25.*Y(5))/
```

```
      1(1F.*DELTAX)
       DYDX=DYDX/5.0
       IF(K.EQ.1)   DRHODH=DYDX
       IF(K.EQ.2)   DVSDH=DYDX
       IF(K.EQ.3)   DCLADM=DYDX
    2  IF(K.EQ.4)   DCDODM=DYDX
C
C    THE PARTIAL DERIVATIVES OF LIFT AND DRAG W.R.T. X1 AND X3
C
       DDDV=Q*(DCDODM/VS+U**2*DCLADM/VS)+RHO*X1*SAPPA*DRAG/Q
       DLVDV=Q*(CLA*U/X1**2+U*DCLADM/(VS*X1))
       DLDH=Q*(CLA*U*DRHODH/(X1*RHO)-U*DCLADM*DVSDH/VS**2)
       DDDH=Q*(-X1*DCDODM*DVSDH/VS**2-U**2*DCLADM*DVSDH*X1/VS**2)+
      1DRAG*DRHODH/RHO
C
C    THE DIFFERENTIAL EQUATIONS OF THE STATE AND ADJOINT-STATE VARIABLES
C
       DXDT(1)=THRUST*COS(U)/XMASS-DRAG/XMASS-GRAVTY*SIN(X2)
       DXDT(2)=XLIFT/(XMASS*X1)+THRUST*SIN(U)/(XMASS*X1)-GRAVTY*COS(X2)
      1/X1
       DXDT(3)=X1*SIN(X2)
       DXDT(4)=X1*COS(X2)
       DXDT(5)=X5*DDDV/XMASS-X6*(DLVDV/XMASS-THRUST*SIN(U)/(XMASS*X1**2)
      1+GRAVTY*COS(X2)/X1**2)-X7*SIN(X2)-X8*COS(X2)
       DXDT(6)=X5*GRAVTY*COS(X2)-X6*GRAVTY*SIN(X2)/X1-X7*X1*COS(X2) +
      1X8*X1*SIN(X2)
       DXDT(7)=(X5*DDDH-X6*DLDH/X1)/XMASS
       DXDT(8)=0.0
       RETURN
       END
$IBFTC FMAS    DECK
       FUNCTION FMASS(TIM)
       GRAVTY=32.2
C
C    MASS AS A FUNCTION OF TIME AND GRAVITY
C
       IF(0.0.LE.TIM .AND.TIM .LT.19.44) FMASS=(1734.-51.25*TIM )/GRAVTY
       IF(19.44.LE.TIM .AND.TIM .LE.28.4 ) FMASS=737.7/GRAVTY
      1-(1.60347E-3*TIM **8-3.21277E-1*TIM **7+29.1634*TIM **6-1.472139
      2E3*TIM **5+4.61362E4*TIM **4-9.19133E5*TIM **3+1.136781E7*TIM **2
      3-7.977601E7*TIM )/(240.*GRAVTY)  -1.01276465E6/GRAVTY
       IF(TIM .GT.28.4) FMASS=512.10246/GRAVTY
       FMASS=FMASS+1266./GRAVTY
       RETURN
       END
$IBFTC FTHRU
       FUNCTION FTHRUS (TIM)
C
C    THRUST AS A FUNCTION OF TIME
C
       IF(TIM .LT.19.44) FTHRUS=1.23E4
       IF(TIM.GE.19.44.AND.TIM.LE.30.35)FTHRUS=1.28275E-2*TIM **7
      1-2.295845*TIM **6+1.749505E2*TIM **5-7.360599E3*TIM **4+1.845449E5
      2*TIM **3-2.757489E6*TIM **2+2.273562E7*TIM -7.977601E7
       IF(TIM .GT.30.35) FTHRUS=0.0
       RETURN
       END
```

```
$IBFTC FRH
      FUNCTION FRHO(X3)
C
C   RHO  AS  A  FUNCTION  OF  X3
C
      IF(X3.LT.0.0)  FRHO=-4.800153E-1[illegible]*X[illegible]3+[illegible].[illegible]00927E-13*X3**2
     1-6.960857E-3*X3+2.376899E-3
      IF(X3  .LT..35332E5) FRHO=.2378E-2 (1.0-.[illegible]E-5*X3  )**4.256
      IF(X3  .EQ..35332E5) FRHO= .727E-3
      TEMP=1.432 + (X3  -.35332E5)/.204[illegible]E[illegible]
      IF(X3  .GT..35332E5) FRHO= (1.02*.[illegible]E-3)/EXP(TEMP)
      RETURN
      END
$IBFTC FVOS
      FUNCTION FVS(X3)
C
C   VS  AS  A  FUNCTION  OF  X3
C
      IF(X3.LT.0.0)  FVS=8.888854E-13*X3[illegible]+1.771[illegible]77E-8*X3**2
     1-3.649881E-3*X3+1.116809E3
      IF(0..LE.X3  .AND.X3  .LE.4.E4) FVS =[illegible].[illegible]7-9[illegible]E-17*X3  **4
     1-2.378208E-12*X3  **3+3.014393E-[illegible]*X3  **2-[illegible].01760[illegible]E-3*X3
     2+1.116911E3
      IF(4.0E4.LT.X3  .AND.X3  .LT.6.0E4) FVS =[illegible].[illegible]47E2
      IF(6.E4.LE.X3  .AND.X3  .LT.1.E5)  FVS =2.[illegible]3[illegible]1E-22*X3  **5
     1-[illegible].832145E-17*X3  **4+6.54[illegible]108E-1[illegible]*X[illegible]  **3-[illegible].04[illegible]5[illegible]6E-7*X3  **2
     2-2.147739E-3*X3  +1.116329E3
      IF(X3  .EQ.1.E5)  FVS =1.0045E3
      IF(1.E5.LT.X3  .AND.X3  .LE.2.E5) FVS =1.[illegible]5[illegible]6E-18*X3  **4
     1-1.238934E-12*X3  **3+3.20960[illegible]E-7*X3  **2-3.17329E-2*X3  +
     22.057099E3
      IF(1.6E5.LT.X3  .AND.X3  .LT.1.7E5)  FVS =1.1057E3
      IF(X3.GT.2.E5)  FVS=1038.
      RETURN
      END
$IBFTC FCL
      FUNCTION FCLA(XM)
C
C   CLA  AS  A  FUNCTION  OF  MACH  NUMBER
C
      IF(0..LE.XM  .AND.XM  .LE.1.0)  FCLA=1.03E-1
      IF(1.0.LT.XM  .AND.XM  .LT.1.6) FCLA=-5.733715E-2*XM  **4
     1+1.736429E-1*XM  **3-1.468177E-1*XM  **2+3.511789E-2*XM
     2+1.025505E-1
      IF(1.6.LE.XM  .AND.XM  .LE.6.0)  FCLA=2.401625E-4*XM  **4
     1+4.54[illegible]6[illegible]E-3*XM  **3-2.637706E-2*XM  **2+[illegible].297798E-2*XM  +
     29.993017E-2
      FCLA=FCLA*57.2958
      RETURN
      END
$IBFTC FCD
      FUNCTION FCDO(XM)
C
C   CDO  AS  A  FUNCTION  OF  MACH  NUMBER
C
      IF(0.0.LE.XM  .AND.XM  .LE.0.6) FCDO =.182
      IF(0.6.LT.XM  .AND.XM  .LE.1.15) FCDO =-11.[illegible]3[illegible]093*XM  **4
```

```
     1+39.7[illegible]C7*XM**3+46.0344[illegible]XM**2[illegible]24.01167*XM-4.492402
      IF(1.15.LT.XM)  FCDO=5.75129[illegible]E-2*XM**4
     1-7.159816E-2*XM**3+3.3757[illegible]E-[illegible]*XM**2-7.710667E-1*XM
     2+0.9904876
      IF(4.472.LT.XM)  FCDO=0.194
      RETURN
      END
$IBFTC INTE
      SUBROUTINE INTEG(TIME,X,CONTR)
      COMMON/TOO/ DTIME,NSTEP,NSTATE,[illegible]MA,COST,FTIM,FCONTR,XVOST,RO,TI
      DIMENSION X(8,100),CONTR(100),TIME(100),F(10,4),TX(10),DXDT(10)
      M=1
      NEG=0
      STORE=1.0
      NSTEP1=NSTEP+1
      DO 14 IND=1,2
      DO 10 JJ=1,NSTEP
      IF(IND.EQ.1.AND.STORE.LT.0.0)  GO TO 25
C
C     COMPUTING   RUNGE-KUTTA   FORMULA
C     X(N+1) = X(N) + (K-0 + 2*K-1 + 2*K-2 + K-3)/6
C
      IF(IND.EQ.1) J=JJ
      IF(IND.EQ.1) J1=JJ+1
      IF(IND.EQ.1) H=DTIME
      IF(IND.EQ.2) J=NSTEP1-(JJ-1)
      IF(IND.EQ.2) J1=NSTEP1-JJ
      IF(IND.EQ.2) H=TIME(J1)-TIME(J)
C
C     FINDING   K-0
C
      FTIME=TIME(J)
      DO 1 K=1,NSTATE
    1 TX(K)=X(K,J)
      CONTRO=CONTR(J)
      CALL EQUAT(TX,DXDT,FTIME,CONTRO)
      DO 2 K=M,NSTATE
    2 F(K,1)=DXDT(K)*H
C
C     FINDING  K-1
C
      FTIME=TIME(J)+0.5*H
      IF(IND.EQ.1) J2=J+1
      IF(IND.EQ.2) J2=J-1
      CONTRO=CONTR(J2)+(FTIME-TIME(J2))*(CONTR(J)-CONTR(J2))/(TIME(J)
     1-TIME(J2))
      DO 3 K=1,NSTATE
    3 TX(K)=X(K,J)+0.5*F(K,1)
      CALL EQUAT(TX,DXDT,FTIME,CONTRO)
      DO 4 K=M,NSTATE
    4 F(K,2)=DXDT(K)*H
C
C     FINDING  K-2
C
      DO 5 K=1,NSTATE
    5 TX(K)=X(K,J)+0.5*F(K,2)
      CALL EQUAT(TX,DXDT,FTIME,CONTRO)
```

```
      DO 6 K=M,NSTATE
6     F(K,3)=DXDT(K)*H
C
C   FINDING  K-3
C
      FTIME=TIME(J) + H
      IF(IND.EQ.1)  CONTRO=CONTR(J+1)
      IF(IND.EQ.2)  CONTRO=CONTR(J-1)
      DO 7 K=1,NSTATE
7     TX(K)=X(K,J)+F(K,3)
      CALL EQUAT(TX,DXDT,FTIME,CONTRO)
      DO 8 K=M,NSTATE
8     F(K,4)=DXDT(K)*H
C
C   FINDING  X(N+1)
C
      DO 9 K=M,NSTATE
9     X(K,J1)=X(K,J)+(F(K,1)+2.0*F(K,2)+2.0*F(K,3)+F(K,4))/6.0
      STORE=X(3,J1)
10    CONTINUE
      GO TO (15,14),IND
25    DO 26 I=1,NSTATE
      DO 26 J=JJ,NSTEP
26    X(I,J+1)=X(I,J)
C
C   EVALUATING  THE  APPROXIMATE  FINAL  TIME
C
15    DO 16 J1=2,NSTEP1
      IF(X(3,J1).LE.0.0) GO TO 17
16    CONTINUE
      WRITE(6,100)
100   FORMAT(//29H INCREASE TIME OF INTEGRATION//)
      TNS=NSTEP
      FTIM=TNS*DTIME
      COST=X(4,NSTEP1)
      FCONTR=CONTR(NSTEP1)
      GO TO 22
17    FTIM=TIME(J1)-(TIME(J1-1)-TIME(J1 ))*X(3,J1)/
     1(X(3,J1-1)-X(3,J1))
C
C  APPROXIMATING   COST   AND  CONTROL  AT  FINAL  TIME
C
      I=4
      COST    =X(I,J1-1)+(X(I,J1+1)-X(I,J1-1))*(FTIM-TIME(J1-1))/
     1(TIME(J1+1) - TIME(J1-1))
      FCONTR    =CONTR(J1-1)+(CONTR(J1+1)-CONTR(J1-1))*(FTIM-TIME(J1-1))/
     1(TIME(J1+1)-TIME(J1-1))
     1(J1-1))
      FCONTR=CONTR(J1-1)+(CONTR(J1)-CONTR(J1-1))*(FTIM-TIME(J1-1))/(TIME
      CONTR(J1)=FCONTR
      NSTEP1=J1
      NSTEP=NSTEP1-1
22    M=NSTATE+1
      NSTATE=2*NSTATE
      N3=NSTEP+3
      X(3,N3)=1.0
      DO 12 I=M,NSTATE
```

```
12    X(I,NSTEP1)=X(I,N3)
14    CONTINUE
      NSTATE=NSTATE/2
      XVOST=PRICE(CONTR)
      COST=0.5*COST**2-XVOST
      DO 30 I=1,NSTEP1,4
30    WRITE(6,103) TIME(I),X(1,I),X(2,I),X(3,I),X(4,I),CONTR(I)
     1,X(5,I),X(6,I),X(7,I),X(8,I)
103   FORMAT(10(E10.3,2X))
      RETURN
      END
$IBFTC PRIC     DECK
      FUNCTION PRICE(CONTR)
      COMMON/TOO/ DTIME,NSTEP,NSTATE,SAREA,COST,FTIM,FCONTR,XVOST,R0,TI
      DIMENSION CONTR(100)
      N=TI/DTIME+3.0
      R=R0
      PRICE=0.5*R*CONTR(1)**2
      PRICE=PRICE+2.0*R*CONTR(2)**2
      DO 1 I=3,N,2
      FAC=I-1
      TIM=FAC*DTIME
      IF(TIM.GT.TI)  R=0.0
      IF(TIM.LE.TI)  R=R0
      PRICE=PRICE+R*CONTR(I)**2
      TIM=TIM+DTIME
      IF(TIM.GT.TI)  R=0.0
      IF(TIM.LE.TI)  R=R0
1     PRICE=PRICE+2.0*R*CONTR(I+1)**2
      PRICE=PRICE*DTIME/3.0
      RETURN
      END
```

# Appendix B

## The Drag and Lift Coefficient Equations

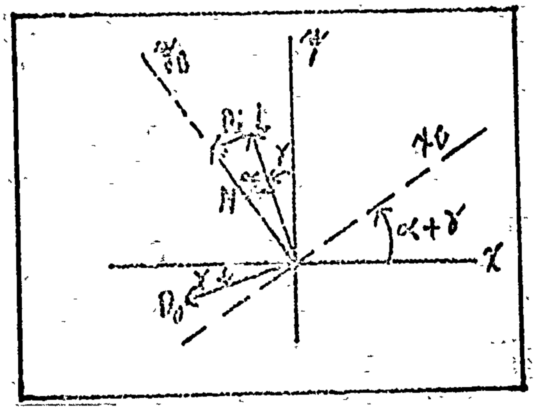


Fig. 1B. The Lift and Drag Forces Acting on the Missile

In Fig. 1B, $x_B$ and $y_B$ are the body axes of the missile. L and N are the lift and normal force vectors, respectively; $D_0$ and $D_i$ are parasite and induced drag, respectively. All other drag components are assumed negligible.

From the diagram it can be seen that

$$D_i = F \sin(\alpha) \qquad (1B)$$

$$L = F \cos(\alpha)$$

If $\alpha$ is small

$$D_i \approx F\alpha$$

$$L \approx F$$

and therefore,

$$D_i \approx L\alpha \qquad (2B)$$

The total drag D can be written as

$$D = D_0 + D_i \qquad (3B)$$

Substituting Eq (2B) into Eq (3B)

$$D = D_0 + L\alpha \qquad (4B)$$

The nondimensional form of Eq (4B) is

$$C_D = C_{D_0} + C_L\alpha \qquad (5B)$$

The lift coefficient can be written as

$$C_L = C_{L\alpha}\alpha \tag{6B}$$

where $C_{L\alpha}$ is the trimmed lift curve slope at a given Mach number. Substituting Eq (6B) into Eq (5B),

$$C_D = C_{D_0} + C_{L\alpha}\alpha^2 \tag{7B}$$

Note that the use of an aerodynamic moment equation is not necessary since only trimmed flight conditions are used to define the lift and drag coefficients.

# Appendix C

## Lagrangian Differentiation Formula

Assume the function F(x) can be represented by the Lagrange interpolation polynomial (Ref 6: 48)

$$f(x) = \sum_{k=0}^{n} l_k(x)\, f_k + E(x) \qquad (1C)$$

where

$$E(x) = \pi(x)\,\frac{f^{(n+1)}(\xi)}{(n+1)!} = \pi(x)\, f[x_0, x_1, \cdots x_n, x]$$

$$l_k(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$$

$$\pi(x) = (x-x_0)(x-x_1)\cdots\cdots(x-x_n)$$

$$f_k = f(x_k)$$

$n+1$ = number of points used in the approximation

$f(x_0, x_1, \ldots x_k, x)$ is the divided difference of the $(k+1)^{th}$ order.

$$f[x_0, x] = \frac{f(x) - f(x_0)}{x - x_0}$$

$$f[x_0, x_1, x] = \frac{f[x_1, x] - f[x_0, x_1]}{x - x_0}$$

$$\vdots \qquad\qquad \vdots$$

$$f[x_0, x_1, \cdots x_k, x] = \frac{f[x_1, \cdots x_k, x] - f[x_0, x_1, \cdots x_k]}{x - x_0}$$

The $x_i$'s ($i=0, 1, \cdots k$) are spaced equally apart.

By differentiating Eq (1C) r times, the $r^{th}$ derivative of f(x) can be approximated as

$$f^{(r)}(x) = \sum_{k=0}^{n} l_k^{(r)}(x)\, f_k + E^{(r)}(x) \tag{2C}$$

Assuming $E^{(r)}(x)$ is small and therefore can be neglected, Eq (2C) becomes

$$f^{(r)}(x) = \sum_{k=0}^{n} l_k^{(r)}(x)\, f_k \tag{3C}$$

which is the general form of the Lagrangian differentiation formula with the error term neglected. To obtain the five formulas used in this paper, set r = 1, n = 4, and $x = x_i$ ($i = 0,1,\ldots 4$) in Eq (3C). These formulas are located in the "Equat" computer subprogram (Appendix A).

## Appendix D

### Polynomial Least Squares Curve Fitting

The Legendre's principle of least squares: given exact or equally reliable data, assume that the best approximation of curve fitting is one for which the aggregate of the squared error over the entire domain is least (Ref 5: 63).

The exact values of f(x) are known at discrete points corresponding to $x_o$, $x_1$, ....$x_m$ over the interval ($x_o$, $x_m$). It is desired to approximate f(x) in the form

$$f(x_i) = \sum_{k=0}^{n} a_k x_i^k \quad (i = 0, 1, \cdots m); \; n \leq m \tag{1D}$$

Define the error r(x) as

$$r(x_i) = f(x_i) - \sum_{k=0}^{n} a_k x_i^k \tag{2D}$$

If the a's are determined such that

$$R = \sum_{i=0}^{m} r^2(x_i) = \sum_{i=0}^{m} \left[ f(x_i) - \sum_{k=0}^{n} a_k x_i^k \right]^2 \tag{3D}$$

is minimum, then the best approximation in the least squares sense is obtained. The minimum of R can be obtained by ordinary calculus:

$$\frac{\partial R}{\partial a_j} = \frac{\partial}{\partial a_j}\left[\sum_{i=0}^{m} r^2(x_i)\right] = 0; \quad (j = 0, 1, \cdots n) \tag{5D}$$

Eq (5D) represents the normal equations. Substituting the a's obtained from Eq (5D) into Eq (10) yields the fitted polynomial.

To increase the accuracy of the curve fit, the range of the independent variable is normalized over $-1 \leq x_i \leq 1$ by applying Eq (6D),

$$x_i' = \frac{2x_i - x_m - x_0}{x_m - x_0} \tag{6D}$$

The following equation yields the coefficients associated with the unnormalized independent variable (Ref 5: 69).

$$a_j = \sum_{i=0}^{n-j} (-1)^i (2^j)\, a'_{j+i} \binom{j+i}{i} y^i k^{j+i} \tag{7D}$$

where

$n =$ degree of the polynominal

$$k = \frac{1}{x_m - x_0}$$

$$\binom{n}{i} = \frac{n(n+1)\cdots\cdots(n-i+1)}{i(i-1)\cdots\cdots 2\cdot 1}$$

$$\binom{n}{1} = n; \quad \binom{n}{0} = 1; \quad \binom{i}{i} = 1$$

$$y = x_0 + x_i$$

## Vita

Albert Ivory Chatmon, son of Saxfield and Emily K. Chatmon, was born on 12 October 1946, in Washington, D.C. In June 1964 he graduated from McKinley High School, Washington, D.C. where he studied in the pre-college engineering curriculum. In June 1968 he graduated from Howard University, Washington, D.C. as a distinguished R.O.T.C. graduate receiving the degree of Bachelor of Science in Electrical Engineering. He was commissioned a Second Lieutenant in the Regular U.S.A.F. He is a student member of the American Institute of Aeronautics and Astronautics and the Institute of Electrical and Electronics Engineers, an E.I.T. member of the Ohio Society of Professional Engineers, and a member of the Air Force Association.

Permanent mailing address: 1758 Kilbourne Place, N.W.
Washington, D.C. 20010

Unclassified

Security Classification

**DOCUMENT CONTROL DATA - R & D**

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

1. ORIGINATING ACTIVITY *(Corporate author)*
Air Force Institute of Technology (AFIT-SE)
Wright-Patterson AFB, Ohio 45433

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE

Range Maximization of an Air-to-Surface Missile

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
AFIT Thesis

5. AUTHOR(S) *(First name, middle initial, last name)*

Albert I. Chatmon

6. REPORT DATE
June 1970

7a. TOTAL NO. OF PAGES

7b. NO. OF REFS
10

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
AFIT Thesis GGC/EE/70-5

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
N/A

10. DISTRIBUTION STATEMENT

[redacted]

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Aeronautical Systems Division
Wright-Patterson AFB, (ASBEI)
Ohio 45433

13. ABSTRACT

Pontryagin's Maximum Principle, coupled with the conjugate gradient iterative technique, is employed in determining estimates of the two-dimensional, maximum range trajectory of an air-to-surface missile. Angle of attack is used as the control parameter.

The motion of the vehicle is described by four state equations including standard atmospheric data, and lift and drag data obtained from wind tunnel tests. In the adjoint equations Lagrangian differentiation formulas are used to approximate the derivatives of lift and drag with respect to velocity and altitude.

Two quadratic cost functions are investigated -- one involving a linear range term and the other a quadratic term. Both include a quadratic penalty function involving a weighting function and the square of the control.

DD FORM 1 NOV 65 1473

Security Classification

Security Classification

| 14. KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| Air-to-Surface Missile Trajectory | | | | | | |
| Conjugate Gradient Method | | | | | | |
| Maximum Range Trajectory | | | | | | |
| Optimal Control | | | | | | |
| Trajectories | | | | | | |
| Angle of Attack | | | | | | |

Security Classification